The Thesis Committee for Namrata Unnikrishnan Nayar
certifies that this is the approved version of the following dissertation:

Model-based fault detection and diagnosis of BLDC motors working at variable speed using wavelet transform

APPROVED BY:

SUPERVISING COMMITTEE:

Raul G. Longoria, Supervisor

Dragan Djurdjanovic

Model-based fault detection and diagnosis of BLDC motors working at variable speed using wavelet transform

by

Namrata Unnikrishnan Nayar

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2019

Acknowledgments

I would firstly like to thank my advisor, Dr. Raul G Longoria for his whole-hearted guidance, assistance, and encouragement throughout my work on this thesis. He has always been available through emails, and regular meetings. Also being understanding and encouraging whenever it was needed. I have learned a lot from his years of experience in modeling dynamic systems through the extremely intuitive method of bond graphs. Him recommending me to pursue an internship at KUKA has helped me find a practical application for the method developed in this thesis. Most importantly, he has been extremely supportive and open to all my ideas.

Not to forget, I would like to thank the brilliant faculty at UT Austin, especially Professors Richard R. Neptune, Maruthi R. Akella and Glenn Y. Masada. I have learned a lot from them. Their unmatched enthusiasm towards conveying concepts in their subjects has contributed immensely to my understanding. The skills I have picked up through the well-designed course projects that they assigned have all shown up in my thesis. Apart from the faculty I have learnt from, I would like to thank Dr. Dragan Djurdjanovic, to find time and give his valuable inputs to my thesis.

I would also like to thank my parents, who are the reason I am here. Their unconditional support and unquestionable trust in my potential have always helped me sail through unpleasant times. Qualities like punctuality, patience, and determination, required to compile the thesis, are traits I solely derive from them.

Lastly, I would like to thank my friends who have been with me through thick in thin. They helped me maintain a good work life balance. The endless debates on interesting topics, treks with soulful music and philosophical discussions have helped me get through uncertain times.

Model-based fault detection and diagnosis of BLDC motors working at variable speed using wavelet transform

Namrata Unnikrishnan Nayar, MSE
The University of Texas at Austin, 2019

Supervisor: Raul G. Longoria

Predictive Maintenance is becoming increasingly important in the automated industry, as maintaining equipment health is essential for the smooth flow of a manufacturing process. This thesis focuses on introducing a model-based technique for the predictive maintenance of robotic motors. The outcomes of this technique, when applied to practical scenarios as well as in simulations, are shown and discussed below.

The thesis aims to provide a strong theoretical justification to an idea, that was developed and tested successfully on real-life robot motor data. The idea involves applying the wavelet transform to the motor current and gathering the occurrence of frequencies relative to the central frequency (frequency ratio) to detect the presence of an anomalous frequency ratio. The enormous amount of information

documented on experiments using Motor Current Signature Analysis (MCSA) for fault detection and diagnosis (FDD) is used here to diagnose the fault.

Simulations of two faulty models, Stator inter-turn winding short and eccentricity fault, have been developed and tested, at variable speeds, with this technique to provide assertive results. Also, the positive results obtained while the technique was applied to the robotic motor data has been presented and explained. This idea is different and more useful that MCSA alone as it works for variable speed conditions. Lastly, the thesis provides suggestions on how to expand on the technique to develop a useful predictive maintenance tool for robotic motors.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Need for predictive maintenance

In leading automobile industries, timely delivery of ordered products and services are crucial for customer satisfaction. The ability to deliver depends largely on the robustness and effectiveness of the assembly line equipment. As a result, staying on top of maintenance schedules and achieving zero down time (ZDT) is a well-established challenge. Predictive maintenance (PM) is an emerging industrial technique that has made ZDT possible. It is especially useful in automobile manufacturing where car bodies come down every 60 to 90 seconds and a downtime can cost up to $20,000 a minute[1].

The method involves predicting equipment failure by analyzing signals coming from sensors attached to the system. Anticipating a failure beforehand helps in two ways. Firstly, to perform adequate maintenance and avoid the occurrence of a failure. And secondly, to acquire a replacement in advance and perform maintenance during a scheduled downtime. In this technique, data from sensors are collected and sent to a central repository. This repository contains system information that provides context to the data gathered.

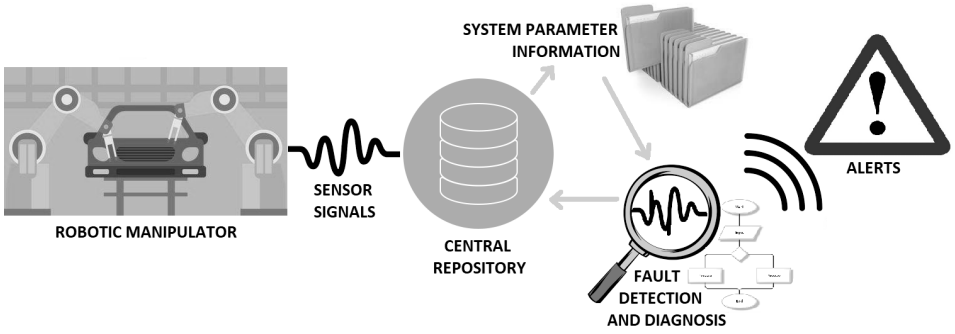


Figure 1.1: Predictive Maintenance

A variation in the trend of the data signal is mostly treated as an anomaly. Algorithms use either system parameter knowledge or the equipment's data history to gauge the machine's health. This idea of implementing PM for fault detection and diagnosis has many merits. It reduces maintenance, increases efficiency and diminishes cost for spare parts and supplies. But it comes at the cost of an expensive installation procedure and hiring a experienced data scientist who can interpret the results.The next section focuses on the faults that normally occur in robotic manipulators.

1.2 Faults occurring in robotic manipulators

A robotic manipulator is basically a set of hardware components and software programs, interacting and working together as a system. They coordinate to complete a required task in the physical world, such as picking and placing an object. As a result, critical faults in the hardware or software can make these sophisticated and expensive devices fail.

Faults are subdivided into hardware, software, and interaction related faults. Hardware faults might occur to any physical component of the robotic system. Such faults influence the information feed of the robot and its ability to perform instruc-

tions. Software faults may regard faulty algorithms and/or faulty implementations of correct algorithms. Such faults influence the cognitive behaviour of the robot, i.e., perception, decision making, and behaviour execution. Interaction related faults are the result of exogenous events like collision, high temperature environment, etc.

Hardware faults in robotic manipulators can be divided into three main categories: sensors, mechanical structure, and actuation system [2]. There are mainly three sensors in a robot: position, velocity and current sensor. The position sensor, an encoder (mostly) may have a fault in the electric board attached to it that relays the reading, hence always giving a constant position error. A fault in the tachometer circuit or acquisition system, almost always causes the control system to read "null velocity". Also a fault in a Hall effect sensor acquisition circuit could lead to a current error.

Mechanical faults in manipulators could be like a blocked joint (due to high friction), a broken belt drive, broken or damaged gear teeth, or even a collision with another object in the workspace. The mechanical drive chains of the axes consist of different standard machinery elements (gears, bearings, toothed belts, shafts, other actuators, etc.) transferring torque from the motor to the robotic arm which are bound to fail due to fatigue, overload, or less lubrication.

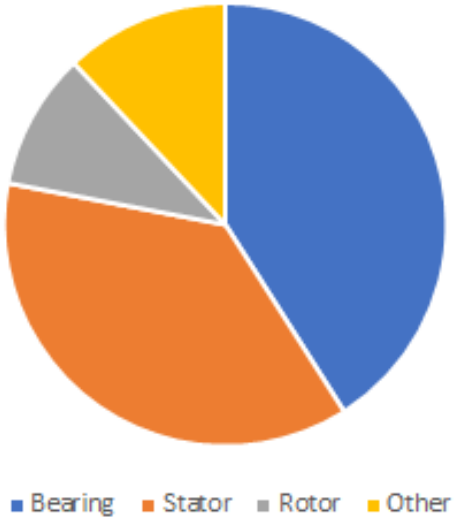
Last, but not the least, an actuation fault includes any faults with the input-output system; i.e., the motor driver: controller which sends input to the motor in order to maintain it at a required speed or torque) and even the motors.

1.3 Robotic motor faults

Motors, common electromechanical actuators used in reobotic systems, are responsible for moving the robotic links. Although aided by pneumatic systems to avoid large loads, motors are the devices that shift around the heavy robotic arms and hence are most susceptible to wear and tear. Motors can be classified into many

types, but those used in the industry need to provide a high ratio of torque to inertia, respond quickly, be resilient, smooth and efficient. Brushless DC motors are normally used as industrial robotic actuators. These motors look like simple and compact machines, but are complex and can fail due to numerous reasons. Some failures are sudden and can't be avoided but some develop over a period of time and can be detected well in advance.

Motor faults are majorly associated with the rotor, stator and bearing. Rotor faults include eccentricity, broken bars, cracked end-rings bent and bent shaft (causing dynamic eccentricity and friction between rotor and stator windings). Stator faults include shorting of one or more of the stator windings due to high temperature or broken rotor bar pieces flying around. Bearing faults involve deformation of the bearing balls or pits in the inner or outer rim of the groove. These deformations can induce rotor faults like eccentricity. Statistically, about 41% of the faults are bearing faults, 37% are stator faults, 10% are rotors faults, and 12% belong to a variety of other faults.[36]



1.4 Methods adopted for fault detection

FDD approaches are typically divided into three categories: data-driven, model-based, and knowledge-based [3]. Data-driven approaches are model free. Online data is usually used to statistically differentiate a potential fault from historically observed normal behaviour, e.g., via Principle Component Analysis. Model-based approaches typically use analytical redundancy to detect and diagnose faults. The correct behaviour of each component in the robotic system would be modelled analytically, and the expected output compared to the observed output. Knowledge-based models involve mathematical equations, which typically describe the functionality of components. Qualitative models involve logic-functions, which typically describe the behaviour of components by describing qualitative relations between the observed variables.

Robotic systems can range from being totally controlled to systems with a high degree of autonomy. A robotic arm in an assembly line, for example, is commonly a fully autonomous robot that works within strict constraints of its direct environment. A system is considered autonomous when it is independent of a human operator, has a remote supporting system, and is able to work for extended periods of time without any intervention. The main difficulty in creating an FDD mechanism for an autonomous robot is generating expectations, quickly, online, with a low computational burden.

One way of carrying out FDD on these robotic systems is through model-driven algorithms. This process explicitly requires a good model of the normal behaviour of the system and also its behaviour in the presence of known faults. These models are a set of analytical equations accompanied by logical operators, used to make decisions. Model-based systems are better than data-driven systems as they can predict the approximate output of the model for any given input, unlike data-driven systems. Inconsistencies between the observed behaviour and the produced

expected behaviour are suspected to be caused by faults. Fault isolation is the process of selecting the (minimal) sets of components, sufficient to distinguish the faults from one another by explaining the inconsistencies. The computational burden depends solely on the type and fidelity of the model.

Statistical data-driven approaches, such as outlier-detection, may not be an attractive choice for an FDD mechanism as large amounts of data are to be processed online, carrying a heavy computational load, and might not be able to detect a fault in time to avoid problems. Yet, some techniques do better than the traditional approaches and address these challenges. Another data-driven approach is to utilize machine learning. Learning produces an FDD model, which can be quickly used online. Learning offline reduces the online computational load, but produces a static model, which may not fit new behaviours like the model-based model. Learning online increases the computational load but produces a dynamic model. An attractive compromise is to add the detected faults to a dynamic model.

Model-based diagnosis (MBD) has the following properties. A system model is provided in terms of components and their interconnections. The component models describe how each component behaves, not how to diagnose them. A domain-independent reasoning engine calculates the diagnoses from the model. The system may have multiple dependent or independent faults. MBD can propose additional measurements to differentiate among diagnoses. MBD does not need any prior computation. Motor current signature analysis is a well known technique used in MBD to detect faults using the motor current's frequency spectrum.

1.5 Importance of stator current for motor FDD

Motor current signature analysis (MCSA) is an important predictive maintenance tool. It qualifies as very good non-invasive fault detection technique. Invasive techniques have implementation difficulties and high cost. MCSA is a relatively

new and rarely utilized technology that is gaining acceptance in industry for its capabilities. Motor current in the three phases of the BLDC motor is responsible for keeping the rotor rotating. So no matter what fault comes up, the current tries to compensate for it. As a result, all changes in the system (fault) leave behind a signature of itself in the current. Using this method can help us detect faults at an early stage and correct them. Hence extending motor life, avoiding a catastrophic breakdown and production outrage. Most of the faults can be picked up by monitoring just one of the three phases of the motor current.

Motor current is sensed by a current sensor. The current signal is then sent through a spectrum analyzer or specialized MCSA instrument. We typically know the profile of a healthy motor current signal and in reality many harmonics are present as well. Faults modulate the motor current signal and give rise to additional sideband harmonics. Faults also result in anomalies in the induced magnetic fields, hence changing the mutual and self inductance of a motor that show up in the motor current frequency spectrum as an abnormal frequency.

1.6 Purpose and Overview of the Thesis

The purpose of this thesis is to provide theoretical understanding and insights into results derived during a study conducted on a project with KUKA in Austin, Texas. The study aimed at developing a technique that could successfully perfor fault detection and diagnosis of robotic motors. MCSA as described in the previous section contributes significantly to the methods explored and developed in this thesis. The usefulness of MCSA is limited to stationary signals. However, robotic motors accelerate and decelerate and hence don't run at a constant speed. Also in the case of fully or partially autonomous robots the routine may differ a bit from one situation to another. As stationarity of measured signals is not guaranteed, MCSA alone is not sufficient to perform effective FDD.

A new approach is applied in this thesis that first implements a wavelet transform and then analyzes frequency ratios (dominant frequencies by central frequencies). The method developed is found to increase the possibility of fault diagnosis by utilizing the system parameters like pole pairs, no of ball bearings, inner and outer race diameter and so on .Several other data-driven methods were found to be capable of detecting but showed no hope in diagnosing a fault. A given data-driven method also always needs a training dataset that is well balanced in the operating conditions it aims to observe and evaluate (faulty and healthy datasets).

As the actuating motor in the robot is assumed to be a brushless BLDC motor, Chapter 2 is dedicated to modelling of this motor and describing how it works. A bond graph model of the BLDC is described. The derivation of the system equations and other important quantities like torque, are presented. The stator current is also an important quantity as that is needed to implement MCSA. Also, the velocity control strategy, normally adopted in industry is explained.

Chapter 3 starts off by giving a brief overview about motor faults. It then dives into the occurrences, reasons, and consequences of a few faults to highlight the importance of having a model for detecting those faults. Further, a dynamic model of a faulty motor model is developed to compare against the normal motor. The model basis is used to describe the two faults covered in this thesis: a) eccentricity, and b) stator inter-turn fault (SITF).

Chapter 4 is reserved to explain a computational code developed in MATLAB. Sufficient detail is provided to allow the reader to reproduce the same outputs and expand this project easily. The code generates plots of important variables, for each of the two faults, with its difference from a healthy motor clearly standing out. Also, this chapter introduces the popular wavelet transform technique and its advantages over Fast Fourier transform (FFT). This is followed by the novel frequency ratio algorithm developed in this thesis. Results of implementing this algorithm on

the current signals coming from the model helps prove the method. The MATLAB Code used is provided in full in the appendix.

As mentioned at the start of this section, the theoretical derivation is a proof for the actual results that where derived from a study initiated at KUKA Austin. The technique developed and explained above worked successfully in detecting faults in robots at KUKA Austin. The method stands out due to its capability to diagnose the type of fault once the system parameters are known. This thesis end by discussing cases at KUKA Austin where this technique worked. Chapter 5 highlights three case studies where the developed algorithm managed to detect the fault. The plots provided are slightly different from those found in previous chapters, but the difference is clearly explained. Unlike most data-driven algorithms, the model based approach can start working without the need of a training dataset. Models using training datasets work well for situation where the end effector follows exactly or almost the same trajectory repeatedly. This idea is versatile in the sense that it works even if the robot end-effector trajectory has changed.

Chapter 6 summarizes the work and covers a few topics that can make this algorithm stronger and useful. It also includes a few suggestions and recommendations for how a company, such as Kuka Austin, can use the methods described to perform predictive maintenance. These methods involve ways by which predictive maintenance can be less computationally expensive and more effective.

Chapter 2

Brushless DC Motor Modeling

2.1 Working Principle and Industrial significance

A BLDC motor construction, removes the need for brushes and commutators making it more resilient.A BLDC motor is powered by DC electricity via an inverter or switching power supply, which produces an AC electric current to drive each phase of the motor via a closed loop controller (utilizing hall effect sensors for feedback). Refer to Fig 2.1.

Like all other motors, BLDC motors also consist of a rotor and a stator [37]. The BLDC motor stator is made from laminated steel stacked up to carry the windings. Windings in a stator can be arranged in two patterns: star pattern (Y) or delta pattern (δ). The major difference between the two patterns is that the Y pattern gives high torque at low RPM and the δ pattern gives low torque at low RPM. We use the Y pattern. We use three pairs of power MOSFETs. Each pair governs the switching on and off of one phase of the motor. The MOSFETs are controlled using pulse-width modulation (PWM) which converts the input DC voltage into a modulated driving voltage. The use of PWM allows the start-up current to be limited and offers precise control over speed and torque. The PWM

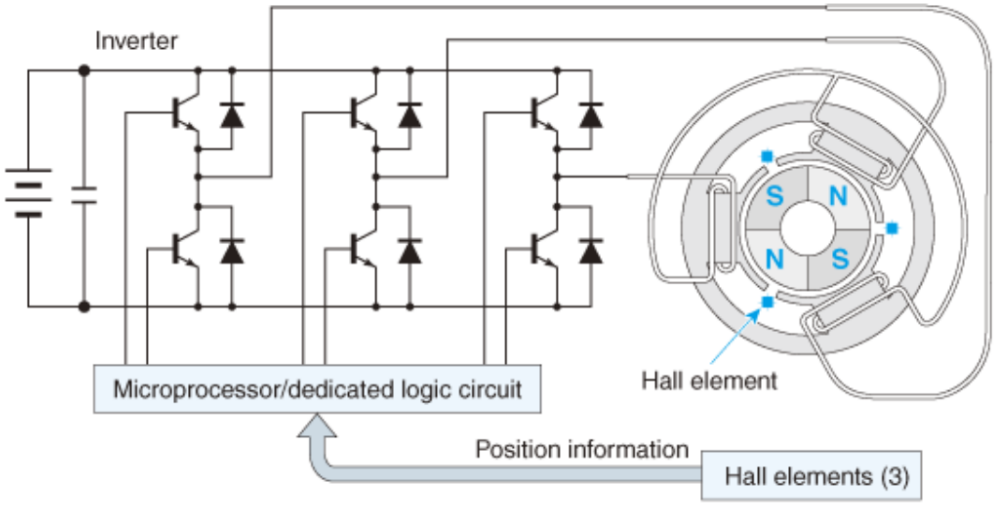


Figure 2.1: Schematic diagram of a permanent magnet BLDC motor[38]

frequency is a trade-off between the switching losses that occur at high frequencies and the ripple currents that occur at low frequencies.

Brushless DC (BLDC) motors have attracted significant interest for applications in robotics because of their large torque-producing capability, high reliability, and low maintenance. Brushless DC motors typically have an efficiency of about 90%, while brushed motors are usually only about 75% efficient. Brushes eventually wear out, sometimes causing dangerous sparking, limiting the lifespan of a brushed motor. Brushless DC motors are quiet, lighter, and have much longer lifespans. Because computers control the electrical current, brushless DC motors can achieve much more precise motion control.

Because of all these advantages, brushless DC motors are often used in continuously running modern devices where low noise and low heat are required. This may include washing machines, air conditioners, and other consumer electronics. These motors are also commonly used in service robots, as careful control of force is required for safety reasons.

2.2 Bond Graph

A bond graph of the BLDC motor is shown in Figure 2.2. The part of the bond graph to the left of the multiport-IC is the electrical side (encapsulating stator dynamics) and the right side is the mechanical side (encapsulating rotor dynamics). The three bonds on the electrical side reflect three coupled inductance effects (the I side), as will be discussed later. The single bond on the C is the mechanical rotational power port. The torque on the bond is the motor torque induced on the rotor.

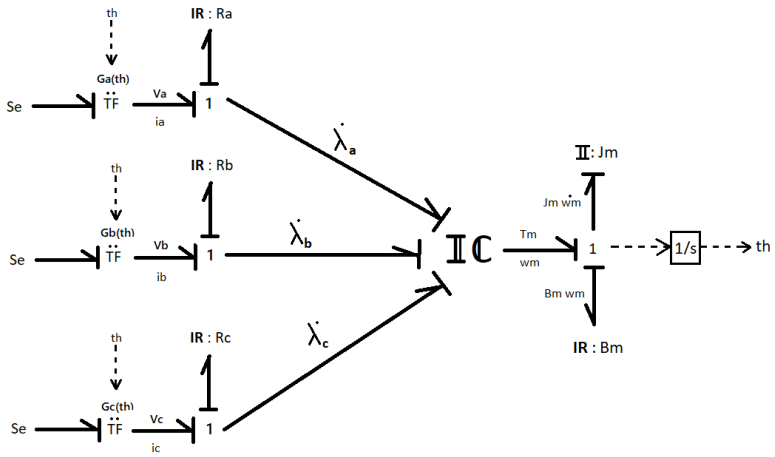


Figure 2.2: Bond graph of Brushless DC Motor

The rotor is assumed to be a permanent magnet. It has been assigned an inertia of J_m, for this model, and is experiences a linear rotational damping of B_m. Coulombic friction will be ignored. The Hall effect sensors are considered to have a good resolution and hence estimate the rotor position well. These angles measurements are sent as a feedback to open and close the gates of the switching circuit to produce three phase-shifted currents.

The effort sources denoted 'Se' are voltage sources that supply three drive voltages that are 120 degrees out of phase. The modulated transformers (TF) use

the rotor position to model the conversion of the input voltages through PWM and switching. The winding is assumed to have a resistance of R_a, R_b, and R_c, respectively. The windings are coupled to the mechanical rotot dynamics through an IC-multiport [11], since the windings have mutual induction. We will notice that the resulting dynamic equations have a coupled inductance matrix.

Summary of key assumptions:

1) The three phase windings are identical

2) The pulse-width modulation, i.e., cyclic switching (on and off of the source) to modulate input power, can be modeled by a modulated transformer (essentially an averaging effect)

3) Heat losses is inherent to the phase coil resistances, but the thermal effects and heat transfer are not modeled in this thesis

4) Slotting effect is not included

2.3 Velocity Control

Proportional and Integral (PI) control is widely used in industrial applications due to its ease in design and simple structure [18, 22]. In application to the BLDC motor, a measurement of the rotor speed is compared with the reference speed and the resulting error is used as an input to the controller at each iteration. BLDC motors are controlled using feedback from the Hall effect sensors, and each electrical cycle has six states. The Hall sensor positions are used for motor commutation, so the drive voltages take the form of the sequences shown in Figure 2.3.

The causality on that bond graph of Figure 2.2 indicates we have four dynamic states, three are the flux linkages of the three phase windings, and one is for the momentum (or angular velocity) of the rotor. We can add an information state for the rotor position. The following sections describe the formulation of the system equations in more detail.

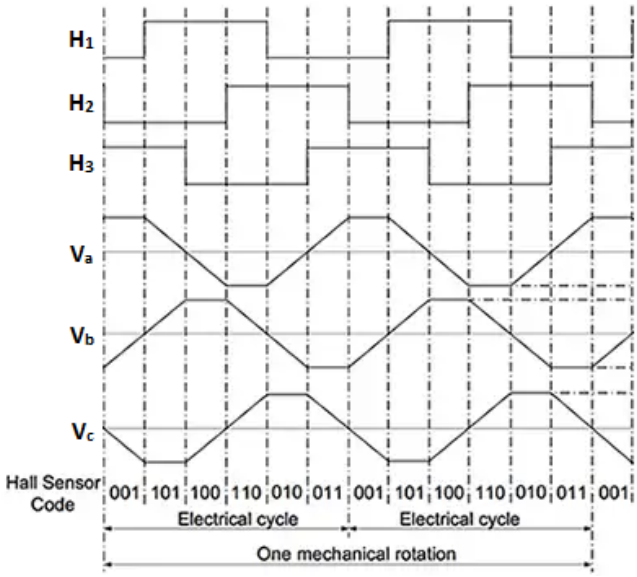


Figure 2.3: Hall effect sensor and voltage output for corresponding mechanical angle

2.4 Voltage equations

We get the three voltage equations given below from the bond graph above by summing the efforts around the three one junctions on the electrical side. Each of these equations is effectively an application of Kirchhoff's voltage law, and gives us one equation for each of the flux linkage states, λ_a, λ_b, and λ_c, or,

$$V_a = R_a i_a + \frac{d\lambda_a}{dt} \tag{2.1}$$

$$V_b = R_b i_b + \frac{d\lambda_b}{dt} \tag{2.2}$$

$$V_c = R_c i_c + \frac{d\lambda_c}{dt} \tag{2.3}$$

where we assume that, $R_a = R_b = R_c = r_s$. The equation given below is the vector equation of the above three equations put together, a form more commonly found

in the electrical machinery literature:

$$\boldsymbol{V_{abc}} = \boldsymbol{R_s i_{abc}} + \frac{d\boldsymbol{\lambda_{abc}}}{dt} \tag{2.4}$$

$$\boldsymbol{R_s} = \begin{bmatrix} r_s & 0 & 0 \\ 0 & r_s & 0 \\ 0 & 0 & r_s \end{bmatrix} \qquad \boldsymbol{\lambda_{abc}} = \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} \tag{2.5}$$

The flux linkage in each winding is due to the self inductance $L_{xx} = L$, mutual inductance $L_{xy} = M$, and linkage due to the interaction between the permanent magnet rotor and stator windings. The constitutive relations between the flux linkages and the phase currents are,

$$\lambda_a = L_{aa} i_a + L_{ab} i_b + L_{ac} i_c + \lambda_{am} \tag{2.6}$$

$$\lambda_b = L_{ba} i_a + L_{bb} i_b + L_{bc} i_c + \lambda_{bm} \tag{2.7}$$

$$\lambda_c = L_{ca} i_a + L_{cb} i_b + L_{cc} i_c + \lambda_{cm} \tag{2.8}$$

where,

$$\boldsymbol{\lambda}_m = \begin{bmatrix} \lambda_{am} \\ \lambda_{bm} \\ \lambda_{cm} \end{bmatrix} = \lambda_m \begin{bmatrix} sin(\theta_e - \frac{2}{3}\pi) \\ sin(\theta_e + \frac{2}{3}\pi) \\ sin\theta_e \end{bmatrix} \tag{2.9}$$

θ_e is the electrical angle

$$\boldsymbol{\lambda_{abc}} = \boldsymbol{L_{ss} i_{abc}} + \boldsymbol{\lambda_m} \tag{2.10}$$

2.5 Torque equations

The IC mulitport element represents the coupled energy stored in the fields, which can be accessible at any of the ports. The three ports on the I side represent how

kinetic energy (electrical) flows into the BLDC, and there are three states, $\lambda_a, \lambda_b, \lambda_c$. The single port on the C side represents how mechanical potential energy is stored, and there is one associated state, θ. In general, for an IC bond graph element:

Flow on I bond : $f = \partial E/\partial p$, and
Efort on C bond : $e = \partial E/\partial q$

Here the $p = \boldsymbol{\lambda_{abc}}$(flux linkage) and $f = \boldsymbol{i_{abc}}$(current). Therefore calculating the total energy E using the flow on I bond equation and (2.6)-(2.10),

$$E = \int \boldsymbol{i_{abc}} \boldsymbol{d\lambda_{abc}} = \int (i_{abc})(\boldsymbol{L_{ss}} d\boldsymbol{i_{abc}} + d\boldsymbol{\lambda_m}) \tag{2.11}$$

giving,

$$\begin{aligned} E = \frac{1}{2}L(i_a^2 + i_b^2 + i_c^2) + M(i_a i_b + i_a i_c + i_b i_c) + \lambda_m i_a sin(\theta_e - \frac{2}{3}\pi) \\ + \lambda_m i_b sin(\theta_e + \frac{2}{3}\pi) + \lambda_m i_c sin\theta_e \end{aligned} \tag{2.12}$$

Now using the effort on C bond equation, where $e = T_m$ and $q = \theta_r$

$$T_m = \frac{\partial E(\boldsymbol{i_{abc}}, \frac{P}{2}\theta_r)}{\partial \theta_r} \tag{2.13}$$

$$T_m = \frac{P}{2}\lambda_m \left[(i_c - \frac{1}{2}i_a - \frac{1}{2}i_b) cos\theta_r + \frac{\sqrt{3}}{2}(i_a - i_b) sin\theta_r \right] \tag{2.14}$$

The above expression denotes the torque acting on the rotor. Next, looking at the mechanical side of equation, at the one junction where $\sum e = 0$,

$$T_m = J_m \dot{\omega}_m + B_m \omega_m \tag{2.15}$$

which provides us with a dynamic equation for the rotor angular velocity, ω_m.

2.6 Summary

This chapter describes the working of a BLDC motor and its relevance to the industry. The bond graph of a BLDC motor is used to understand the working of the system properly and to interpret its dynamics with clarity. This is followed by the discussion of one of the most common control strategies adopted for the motor. Finally we derive the system equations using the bond graph and also derive an expression for the induced rotor torque.

Knowing the model is essential to study the working of a healthy BLDC motor, allowing us to later simulate induced faults. The results derived from the simulation can then be verified with the results obtained from the field, such as the results to be shown in Chapter 5 from KUKA. The next chapter shows how fault models can be introduced into the baseline model just derived.

Chapter 3

Modeling Common Faults in a BLDC Motor

3.1 Major motor faults

Internal faults of a BLDC motor can be categorized as bearing faults, rotor faults, and stator faults. The motor can also experience faults due to external components, such as failures in the Hall effect sensor or gear-box. In the following, we examine some key faults in more detail.

Bearing Faults. A majority of electrical machines use rolling element bearings. Each bearing ball is trapped between two rings called the inner and the outer rings. A set of balls or rolling elements placed in raceways rotate within these rings. Bearing failures are responsible for the highest incidence of recorded motor failures. A continued stress on the bearings leads to fatigue failure eventually, at the inner and/or outer rings of the bearings. Small pieces break loose from the bearing, from a process referred to as flaking or spalling. These failures result in high friction at the bearings that generates detectable vibrations and increased noise levels. This process can also result from other external working conditions, like contamination,

corrosion, improper lubrication, improper installation, and brinelling. These faults can also be the cause of eccentricity faults.

Rotor Faults. In an ideal machine, the rotor is centre-aligned with the stator bore, and the rotor's centre of rotation is the same as the geometric centre of the stator bore. Eccentricity is a condition where there is an unequal air gap between the stator and the rotor. Air-gap eccentricity can be either static or dynamic in form. In the case of a static eccentricity, the air gap remains a constant, invariant of rotor angular displacement. A dynamic eccentricity occurs when the centre of the rotor is not at the centre of rotation and the minimum air gap keeps changing with changes in the rotor angle. This means that a dynamic eccentricity is a function of space and time. We cover this fault in a subsequent section.

An air-gap flux disturbance that results from damaged rotor magnets is also an important fault that occurs frequently in BLDC motors. Deformities formed during manufacturing can lead to more damage and disintegration at high speeds. A chipped magnet may also end up in the air gap, causing increased friction and potential damage to the stator insulation. Flux disturbances may cause unbalanced magnetic pull on the rotor by the stator resulting in more stress on the bearings.

Stator Faults. The most frequently occurring stator fault is the winding insulation breakdown. This usually occurs in the region where the end windings enter the stator slots. It is caused by large electrical voltage stresses, electro-dynamic forces produced by winding currents, thermal aging from multiple heating and cooling cycles, and mechanical vibrations from internal and external sources. This winding insulation breakdown can result in stator inter-turn faults. This fault is discussed more in the following section.

Ancillary Equipment Faults. The loss of a Hall sensor can result in torque pulsations when the rotor is rotating. During such a failure, starting a BLDC motor using only feedback control becomes impossible. Other examples of this type of

fault include shaft coupling misalignments and broken gear teeth.

3.2 Stator winding fault

3.2.1 Explanation of the fault and its relevance

A stator inter-turn fault (SITF) can be classified into three types: between the two coils of the same phase, between the two coils of different phases (turn-to-turn fault), or a turn-to-ground fault. Because the first case is the most common and generally occurs first, it is considered here. We also assume that each phase winding consists of turns that are connected in series and that the three-phase windings are wye-connected.

The heat generated in the short circuit is proportional to the square of the circulating current, I_f, and therefore can cause insulation breakdown in the adjacent coils. Moreover, the propagation of the faults in a single phase could quickly lead to complete failure or shutdown of the motor. However, SITF and complete failure of the motor do not occur at the same time. Thus, fast detection of the SITF during motor operation can eliminate subsequent damage to the adjacent coils and stator core. An early detection technique can help minimize loss of assets and reduce repair costs.

A short circuit (SC) fault in one of the coils induces circulating current in SC turns [15, 19]. The direction of this current is opposite to the windings current and causes an opposing magnetic field. This enhances the magnetic fields in other parts of the motor. Obviously, more faulty turns lead to a larger magnetic field. This distorts the field distribution and the asymmetrical magnetic field can cause saturation in other teeth of the motor. Distortion of the air gap field can directly influence the developed torque. On the other hand, the SC windings current is in opposite direction of the healthy windings current causing a braking torque. This

reduces the mean torque particularly over higher speed.

SITF is a critical problem because this type of fault spreads rapidly to the whole winding. This type of fault accounts for 21% of the faults that occur in electrical machines. It has been reported that most short-circuit faults begin as SITFs, which occur because of insulation breakdown, but develop into more serious faults very quickly. Therefore, an online SITF diagnosis method is required for safe operation and rapid fault reaction.

3.2.2 Faulty model explanation with assumptions and diagram

Suppose the circuit short occurs in the phase B [34], as illustrated in Figure 3.1. Here we attach a resistance R_f to a segment of the winding B. The faulty segment is μ times the entire winding length, where,

$$\mu = \frac{Nsh}{Ns} \tag{3.1}$$

If the resistance R_f is set to a really low number like 0.001 to 0.1 ohms, most of the current i_b will flow through the resistance R_f.

This will result in less useful flux linkage from the μ segment of winding B. The capacity to produce the required torque will decrease drastically. But if we increase the value of R_f to 100 ohms or so, i_f will be smaller and the shorted part wont consume a majority of the current. We can theoretically state that if R_f is zero, The effectiveness of the μ segment of the winding is entirely lost and if R_f is infinity, it means no fault is present.

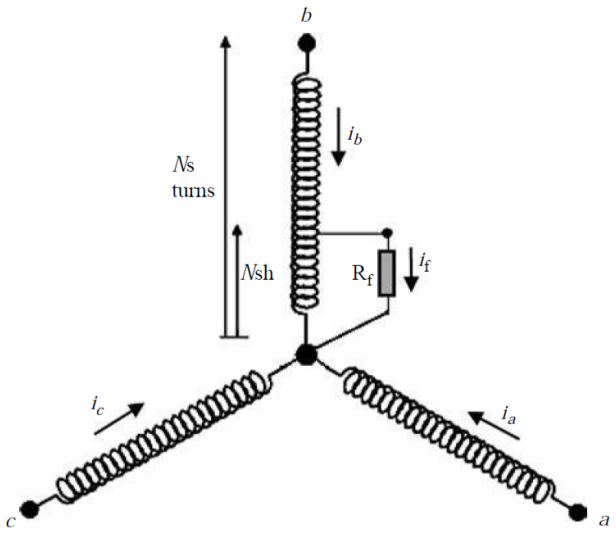


Figure 3.1: Schmatic representation of stator inter-turn fault

3.2.3 Derived Equations

We know that inductance is directly proportional to the numbers of turns and indirectly proportional to the reluctance; i.e.,

$$L_{xx} = \frac{n_x n_x}{R_{xx}}$$

$$M_{xy} = \frac{n_x n_y}{R_{xy}}$$

As a result the self inductance has a squared section of the winding involved, during a fault. Let's first look at the equations of a healthy motor winding, shown in Figure 3.2(bottom).

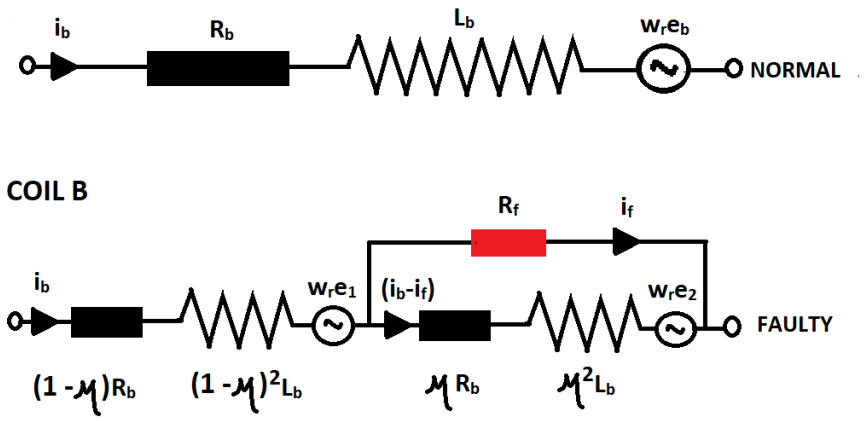


Figure 3.2: Circuit representation of phase B, (top to bottom) Healthy, Faulty with the stator winding shorted by a resistance R_f

In this case, the voltage equations take the form,

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} r_s & 0 & 0 \\ 0 & r_s & 0 \\ 0 & 0 & r_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} \tag{3.2}$$

where $$\boldsymbol{\lambda_s} = [\lambda_a \ \lambda_b \ \lambda_c]^T = \boldsymbol{L_{ss}} \boldsymbol{i_s} + \boldsymbol{\lambda_m}$$

$$\boldsymbol{L_{ss}} = \begin{bmatrix} L & M & M \\ M & L & M \\ M & M & L \end{bmatrix} \qquad \boldsymbol{\lambda}_m = \lambda_m \begin{bmatrix} sin(\theta_r - \frac{2}{3}\pi) \\ sin(\theta_r + \frac{2}{3}\pi) \\ sin(\theta_r) \end{bmatrix} \tag{3.3}$$

In contrast, when there is a stator fault as designated in Figure 3.2 (top), we need an extra state. This can be illustrated by the simplified bond graphs presented in Figure 3.3 Assumption: The stator winding (say B) also experiences an inductance due to the remaining two phase windings (A and C). This component is avoided for simplicity but should not effect the number of states. In the figure given below, it

has been demonstrated how a winding with stator fault requires one extra state to represent its dynamics in entirety.

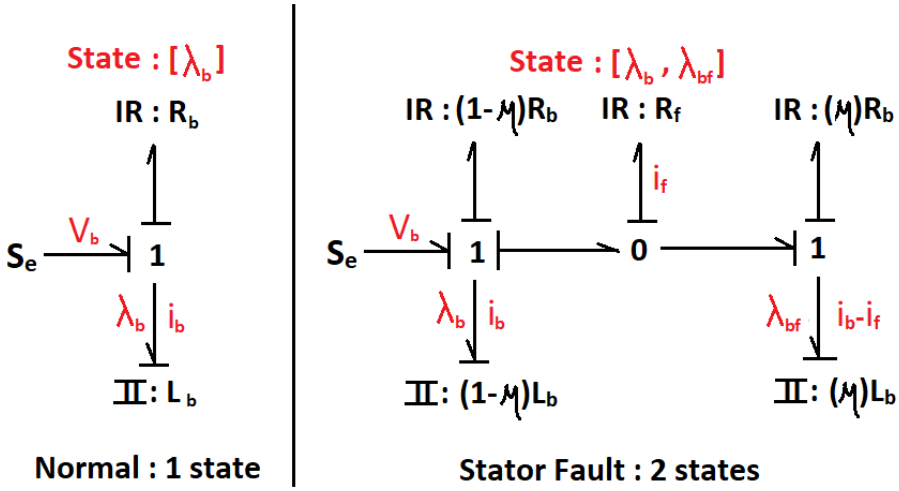


Figure 3.3: Bond Graph of stator winding, (left to right) Healthy and Faulty

Now we derive the full equation using mutual inductances and back emf as well, guided by Figure 3.4. The voltage equations are now,

$$v_1 = R_{b1} i_b + L_{b1} \frac{di_b}{dt} + M_{21} \frac{d(i_b - i_f)}{dt} + M_{a1} \frac{di_a}{dt} + M_{c1} \frac{di_c}{dt} + w_r e_1$$

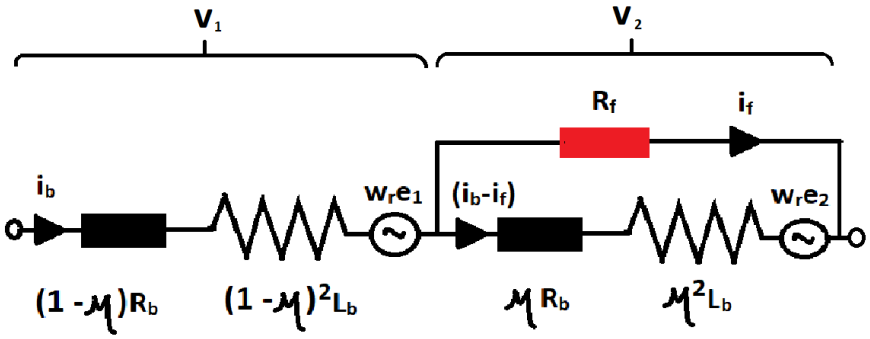


Figure 3.4: Circuit Representation to derive the required voltage equations

$$v_2 = R_{b2}(i_b - i_f) + L_{b2}\frac{d(i_b - i_f)}{dt} + M_{12}\frac{di_b}{dt} + M_{a2}\frac{di_a}{dt} + M_{c2}\frac{di_c}{dt} + w_r e_2$$

with,

$R_b = R_{b1} + R_{b2} = r_s$

$M_{ab} = M_{a1} + M_{a2} = M$

$M_{cb} = M_{c1} + M_{c2} = M$

$L_{bb} = L_{b1} + L_{b2} + M_{12} + M_{21} = (1-\mu)^2 L + \mu^2 L + 2\mu(1-\mu)L = L$

$$v_b = v_1 + v_2 = r_s i_b + M\frac{di_a}{dt} + L\frac{di_b}{dt} + M\frac{di_c}{dt} - \mu r_s i_f - L_{b2}\frac{di_f}{dt} - M_{21}\frac{di_f}{dt} + w_r e_b \quad (3.4)$$

Consequentially,

$$v_a = r_s i_a + L\frac{di_a}{dt} + M\frac{d(i_b - i_f)}{dt} + M\frac{di_c}{dt} + w_r e_a \quad (3.5)$$

$$v_c = r_s i_c + M\frac{di_a}{dt} + M\frac{d(i_b - i_f)}{dt} + L\frac{di_c}{dt} + w_r e_c \quad (3.6)$$

Also, the voltage drop across the shorted part of winding b (the entire loop) is,

$$v_f = 0 = R_f i_f + R_{b2}(i_f - i_b) + L_{b2}\frac{d(i_f - i_b)}{dt} - M_{12}\frac{di_b}{dt} - M_{a2}\frac{di_a}{dt} - M_{c2}\frac{di_c}{dt} + w_r e_f$$

$$v_f = 0 = -R_{b2} i_b + (R_f + R_{b2}) i_f - M_{a2}\frac{di_a}{dt}(M_{12} + L_{b2})\frac{di_b}{dt} - M_{c2}\frac{di_c}{dt} + L_{b2}\frac{di_f}{dt} - w_r e_2$$

$$\begin{aligned} v_f = 0 = &- \mu r_s i_b + (R_f + \mu r_s) i_f - \mu M\frac{di_a}{dt} - \mu M\frac{di_c}{dt} \\ &- (\mu(1-\mu)M + \mu^2 L)\frac{di_b}{dt} + \mu^2 L\frac{di_f}{dt} - \mu w_r e_b \end{aligned} \quad (3.7)$$

Putting equations 3.3 to 3.6 in state space form,

$$\begin{bmatrix} v_a \\ v_b \\ v_c \\ v_f(=0) \end{bmatrix} = \begin{bmatrix} r_s & 0 & 0 & 0 \\ 0 & r_s & 0 & -\mu r_s \\ 0 & 0 & r_s & 0 \\ -\mu r_s & 0 & 0 & \mu r_s + R_f \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_f \end{bmatrix} + w_r \begin{bmatrix} e_a \\ e_b \\ e_c \\ \mu e_b \end{bmatrix}$$
$$\begin{bmatrix} L & M & M & -\mu M \\ M & L & M & -\mu^2 L - \mu(1-\mu)M \\ M & M & L & -\mu M \\ -\mu M & -\mu^2 L - \mu(1-\mu)M & -\mu M & \mu^2 L \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_f \end{bmatrix} \tag{3.8}$$

3.2.4 Frequency Domain

Using the equations (3.2) and (3.8), when the motor is run with and without the SITF fault, variations in the stator currents are recorded. The Fast Fourier Transform (FFT) and Continuous Wavelet Transform (CWT) results for the same are illustrated below in figure 3.5.

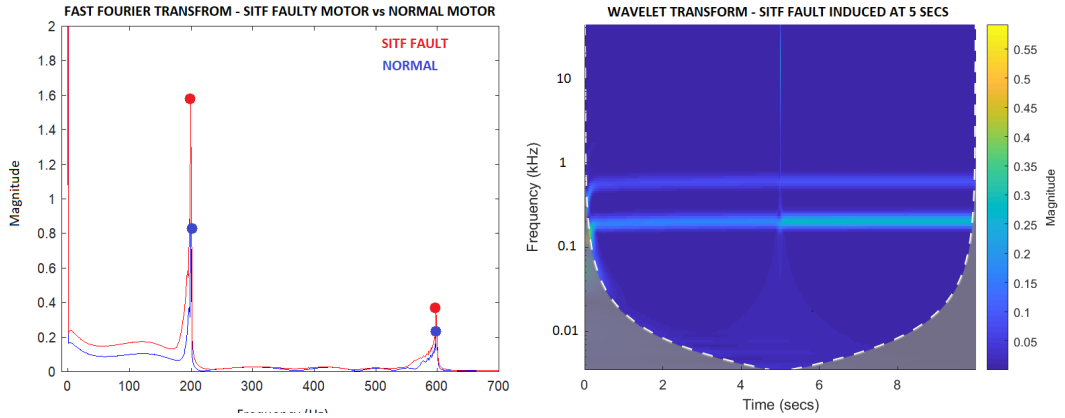


Figure 3.5: Experimental results from model developed in this thesis

In [5] , the current spectrum of a motor with stator inter-turn fault is com-

pared to the current spectrum of a healthy motor. Refer to figure 3.6. They also provide references demonstrating the theory behind the origin of these harmonics. According to the paper the frequency relevant to this fault is given as

$$f_{SITF} = (2n-1)f_c \tag{3.9}$$

where, n = 1,2,3... and f_c = central frequency

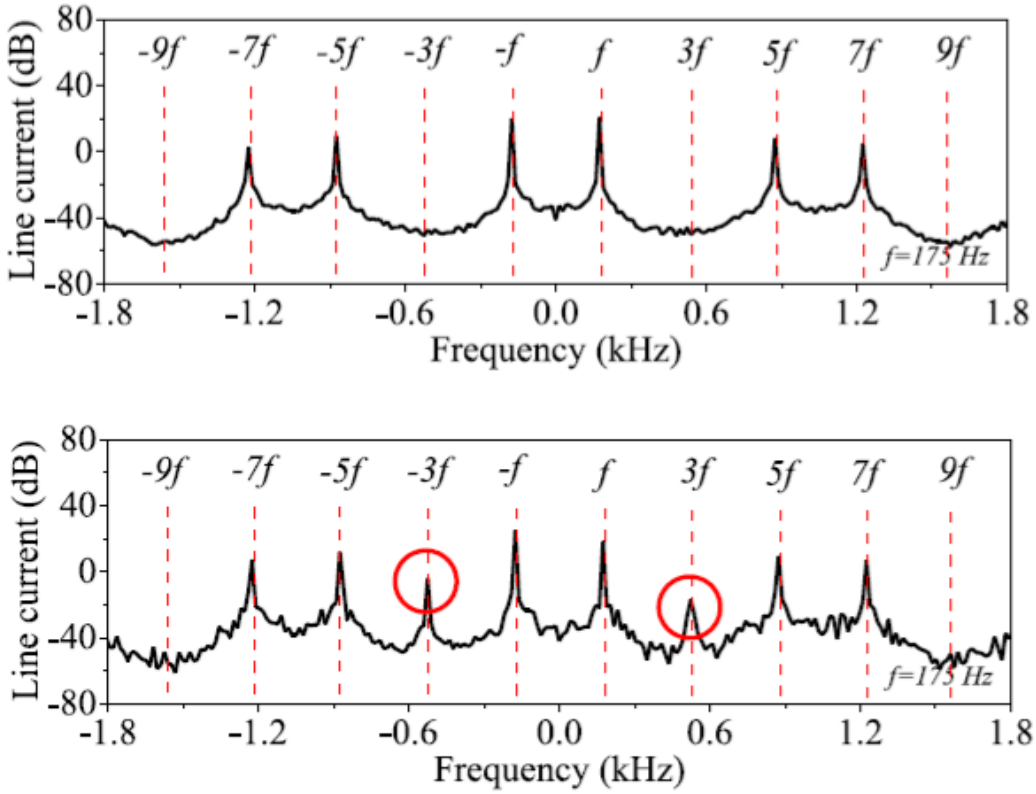


Figure 3.6: Current spectra of phase current (top) Healthy motor characteristics, (bottom) Motor with a stator fault [5]

3.3 Eccentricity fault

3.3.1 Explanation of the fault and its relevance

Static eccentricity in electrical motors occurs when the rotor symmetrical axis is the same as the rotor rotational axis; however, they are dislocated with respect to the stator symmetrical axis. This is illustrated in Figure 3.7. This eccentricity is fixed and, hence, the position, so the air gap function is invariant in time. In this situation, the mutual inductances between the stator and rotor and self- and mutual inductances between the stator phases changes. These inductances are, however, invariant and independent of the rotor angular position, the same as a healthy motor condition.

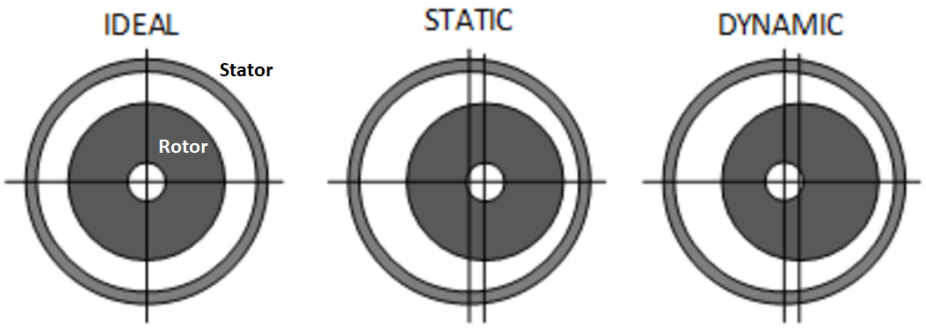


Figure 3.7: Cross-section of BLDC motor with and without eccentricity

A static eccentricity can be due to numerous causes, such as elliptical stator core, wrong placement of the rotor or stator at the setup or subsequent to maintenance, incorrect bearing positioning, bearing deterioration, shaft deflection, housings imperfection, end-shield misalignment, excessive tolerance, rotor weight, or pressure of interlocking ribbon. Accordingly, the static eccentricity leads to additional failures, which results in drastic harm to the rotor, stator core, and windings. The radial forces in a static eccentricity condition produce a steady unbalanced

magnetic pull (UMP) in the radial route across the motor, due to the fact that the reluctance of the magnetic flux path decreases on the side of tiny air-gap.

However, the winding current induces more magnetic flux that causes a stronger pull and leads to expanding the air-gap on the opposite side where the reluctance increases; thus, the flux and magnetic side pull decrease. Therefore, the UMP compels the rotor to move toward the area of the narrowest airgap length until the abrasion of the stator core subsequently generates abnormal vibration and severe damage to the rotor, the windings, and the stator. Consequently, the static eccentricity causes acoustic noise, premature failure in the bearing, rotor deflection, and bent rotor shaft.

Dynamic eccentricity has similar effects. In dynamic eccentricity, the rotor rotational axis doesn't match the rotor symmetric axis but does lie on the stator symmetric axis. As a result the air gap function keeps changing with rotor angle and is time dependent.

3.3.2 Faulty model explanation with assumptions

In this thesis we model static, dynamic but majorly mixed eccentricity [24, 25, 26]. As a result, the air gap between the rotor and the stator is not uniform throughout, and varies with rotor angle. A similar trend is seen in the flux linkage function with angle. The flux due to the permanent magnet is the highest at 0 rads and lowest at π rad. Hence the net change in flux differs from phase to phase. A mixed eccentricity of 20% dynamic and 60% static is studied in simulation in the next chapter. The variation in induction for both static and dynamic eccentricity are individually tracked as eccentricity varies from 0 to 70%.

Derivation of the self and mutual inductances for the eccentric fault case can be very cumbersome, especially when biasing is to be avoided. Consequently, only one such result in derived in this study to demonstrate the approach.

3.3.3 Derived Equations

Since the gap is not constant, the inductance does get affected and is different for different rotor positions. The self inductance and mutual inductance between the winding vary up and down with every rotation, and the flux linkage between the permanent magnet and stator coils also vary up and down due to its dependence on airgap. As regards the dynamic equations, it might intuitively be surmised that an eccentricity will not effect the resistance of the winding.

Despite these intuitive notions, it will be useful to have an air gap function that helps us calculate the air gap at different rotor angles. To illustrate, we review a few simple examples to understand the derivation of motor inductance.

We know that, by Ampere's law, the line integral (circulation) of the magnetic field around some arbitrary closed curve is proportional to the total current enclosed by that curve. As illustrated in Figure 3.8, the integral of magnetic flux strength H along the closed dashed path is equal to N times the current in the coil, by Ampere's law, or,

$$mmf = \int Hdl = Ni \tag{3.10}$$

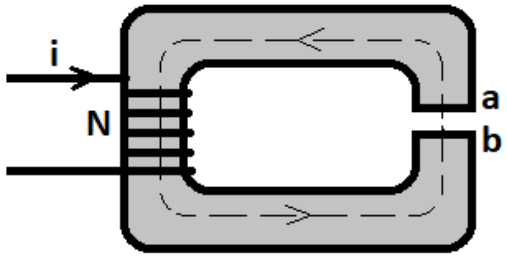


Figure 3.8: Simplified example to understand Ampere's law

The flux passes through iron (a to b) and air (b to a). Also, it is assumed

that the magnetic strength remains constant throughout the material. Therefore,

$$H_i l_i + H_g l_g = Ni \tag{3.11}$$

Assume magnetic flux density B is independent of the permeability of a material, and $B = \mu H$, thus,

$$\frac{B_i}{\mu_i} l_i + \frac{B_g}{\mu_g} l_g = Ni \tag{3.12}$$

Also with the surface integral of flux density equal to the flux, $\phi = \int_A B ds$,

$$\phi_i = B_i A_i \qquad\qquad \phi_g = B_g A_g$$

$$\frac{l_i}{\mu_i A_i}\phi_i + \frac{l_g}{\mu_g A_g}\phi_g = Ni \tag{3.13}$$

One way to think about this is, Ni is the voltage, ϕ is the current and $\frac{l}{\mu A}$ is a resistance. In the magnetic domain, this 'resistance' is called reluctance. Hence,

$$(R_i + R_g)\phi = Ni \tag{3.14}$$

Now, with the concept of reluctance can be used to look at an example that goes over the derivation of inductance of two current carrying coils. In Figure 3.9, the

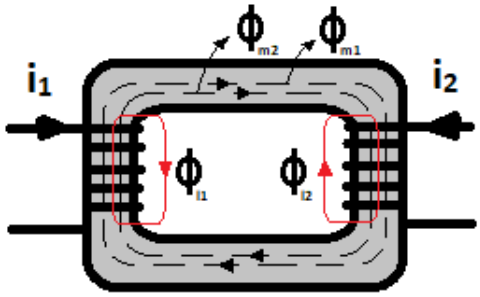


Figure 3.9: Simplified example to understand the idea of leakage and magnetizing flux

flux produced by each winding can be separated into two components - leakage flux, denoted by 'l' and magnetizing flux denoted by 'm'. Hence the flux running through coil 1 and 2 are given below.

$$\phi_1 = \phi_{l1} + \phi_{m1} + \phi m2 \tag{3.15}$$

$$\phi_2 = \phi_{l2} + \phi_{m2} + \phi m1 \tag{3.16}$$

Let's look at ϕ_1, so computing the same for ϕ_2 is similar. From 3.12, $R\phi = Ni$. So,

$$\frac{N_1 i_1}{R_{l1}} + \frac{N_1 i_1}{R_m} + \frac{N_2 i_2}{R_m} \tag{3.17}$$

Finally, as flux linkage λ is equal to the product of the number of turns and flux following through the coil, i.e., $\lambda = N\phi$,

$$\lambda_1 = N_1^2 i_1 (\frac{1}{R_{l1}} + \frac{1}{R_m}) + \frac{N_1 N_2 i_2}{R_m} \tag{3.18}$$

$$\lambda_1 = L_{11} i_1 + L_{12} i_2$$

where, $L_{11} = \frac{N_1^2}{R_{l1}} + \frac{N_1^2}{R_m} = L_{l1} + L_m$ and $L_{12} = \frac{N_1 N_2}{R_m} = L_m$ (if $N_1 = N_2$)

So, in general,

$$L_{xy} = \frac{N_x N_y}{R} = \frac{N_x N_y \mu A}{l} \tag{3.19}$$

Applying the same method to a Brushless DC motor with a 3-phase supply, the inductance can be derived similarly. The only difference is that, while current in these winding are in phase, the currents in a BLDC motor are 120 degrees out of phase. So assuming the ideal case where each phase has been set up perfectly similar to the other two, the inductance works out to be,

$$\begin{bmatrix} L_l + L_m & -\frac{L_m}{2} & -\frac{L_m}{2} \\ -\frac{L_m}{2} & L_l + L_m & -\frac{L_m}{2} \\ -\frac{L_m}{2} & -\frac{L_m}{2} & L_l + L_m \end{bmatrix} \tag{3.20}$$

With all this discussion, it is now possible to derive the equations for the self- and mutual inductances in a PMBLDC motor. As for example one, we can find

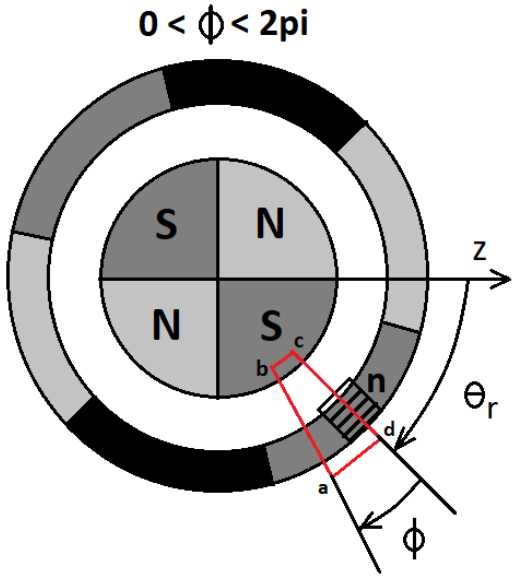


Figure 3.10: Applying Ampere's law to the red coil in the Brushless DC Motor

the MMF around the loop abcda in the figure given above[13]. There is flux flowing along this path due to the rotor magnet and also current in the coils.

$$\int_{abcda} Hdl = ni \tag{3.21}$$

$$F_{ab}(\phi, z, \theta_r) + F_{bc} + F_{cd}(0, 0, \theta_r) + F_{da} = n(\phi, z, \theta_r)i \tag{3.22}$$

F_{bc} & F_{da} are parts of the loop in the iron surface which has a relative permeability μ_{rel} of 4000-5000, unlike air where μ_{rel} is 1. Hence, F_{bc} & F_{da} are negligible. So,

3.19 becomes,

$$F_{ab}(\phi, z, \theta_r) + F_{cd}(0, 0, \theta_r) = n(\phi, z, \theta_r)i \tag{3.23}$$

In the above equation, $n(\phi, z)$ is the number of current carrying conductors the pass through the loop abcda. F is the magneto motive force.

Lets divide the above equation by the air gap and integrate the quotients from 0 to 2π.

$$\int_0^{2\pi} \frac{F_{ab}(\phi, z, \theta_r) + F_{cd}(0, 0, \theta_r)}{g(\phi, \theta_r)} d\phi = i \int_0^{2\pi} \frac{n(\phi, z, \theta_r)}{g(\phi, \theta_r)} d\phi \tag{3.24}$$

Integrating F_{ab} from 0 to 2π is zero as the amount of current going in is equal to the amount of current coming out in an entire rotation. Hence F_{cd} is as below,

$$F_{cd}(0, \theta_r) = \frac{i}{2\pi < g^{-1}(\phi, \theta_r) >} \int_0^{2\pi} n(\phi, \theta_r) g^{-1}(\phi, \theta_r) d\phi \tag{3.25}$$

Substituting 3.23 in 3.21

$$F_{ab}(\phi, z, \theta_r) = n(\phi, z, \theta_r)i - \frac{i}{2\pi < g^{-1}(\phi, \theta_r) >} \int_0^{2\pi} n(\phi, \theta_r) g^{-1}(\phi, \theta_r) d\phi$$

$$F_{ab}(\phi, z, \theta_r) = \left[n(\phi, z, \theta_r) - \frac{1}{2\pi < g^{-1}(\phi, \theta_r) >} \int_0^{2\pi} n(\phi, \theta_r) g^{-1}(\phi, \theta_r) d\phi \right] i$$

$$M(\phi, \theta_r) = \left[n(\phi, z, \theta_r) - \frac{1}{2\pi < g^{-1}(\phi, \theta_r) >} \int_0^{2\pi} n(\phi, \theta_r) g^{-1}(\phi, \theta_r) d\phi \right]$$

$M(\phi, \theta_r)$ is the modified winding function[16, 17]. This is equivalent to number of windings that are within the flux field. Finally, to find the total inductance we need to go an entire rotation and apply equation 3.17.

Here, length is equal to the air gap (look at ab of the loop of abcda) and area A is $2\pi rl$ where l is the stack length of the motor. It is the surface of a cylindrical

surface that has a radius equal to inner stator surface. Hence the inductance is,

$$L_{xy} = \mu r l \int_0^{2\pi} n_x(\phi, \theta_r) N_y(\phi, \theta_r) g^{-1}(\phi, \theta_r) d\phi \tag{3.26}$$

where,

$$N_y(\phi, \theta_r) = n_y(\phi, \theta_r) - \frac{1}{2\pi < g^{-1}(\phi, \theta_r) >} \int_0^{2\pi} n_y(\phi, \theta_r) g^{-1}(\phi, \theta_r) d\phi \tag{3.27}$$

$(x, y) \in (a, b, c, r)$ to give the required self and mutual inductances. When there is no eccentricity, $g^{-1}(\phi, \theta_r) = 1/g_0$. As a result, the self and mutual inductances remain a constant. But when there is dynamic eccentricity the air gap keeps changing and so the self and mutual inductances oscillate.

In this thesis the winding function $n_a(\phi, \theta_r)$ is assumed to take the form shown in Figure 3.11. n_b and n_c are just shifted by 120 degrees. Further trying

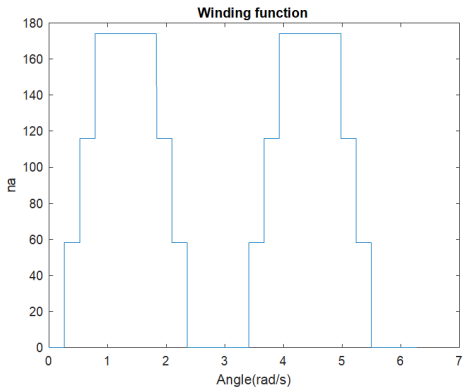


Figure 3.11: The winding function of stator phase A

to derive a function for air gap, As evident from the figure above, the presence of dynamic eccentricity makes air gap a function of rotor position[32]. The degree of dynamic eccentricity δ_d i.e. the quantity lying between 0 and 1 the denotes how

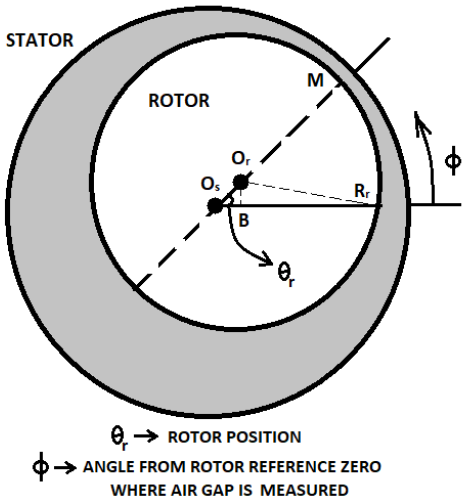


Figure 3.12: Cross-section of motor displaying the variation in air gap with eccentricity

displaced is the rotor, is given by,

$$\delta_d = \frac{|OO'|}{g_0} \tag{3.28}$$

Hence,

$$g(\theta_r, \phi) = g_0 - OM(\theta_r, \phi) \tag{3.29}$$

OM is the distance from stator center to the end of the rotor rim,

$$OM(\theta_r, \phi) = OB + BM$$

Utilizing the Figure 3.12 and equation 3.26,

$$OM(\theta_r, \phi) = \delta_d g_0 cos(\phi - \theta_r) + \sqrt{R_r^2 - \delta_d^2 g_0^2 sin(\phi - \theta_r)^2}$$

The second term is approximately taken as R_r. Also, we know that $R_s = R_r + g_0$. Eq 3.29 turns out to be,

$$g(\theta_r, \phi) = g_0[1 - cos(\phi - \theta_r)]$$

Therefore,inverse air gap function is given by,

$$g^{-1}(\theta_r, \phi) = \frac{1}{g_0[1 - cos(\phi - \theta)]} \tag{3.30}$$

To avoid zero in the denominator, we can express it in Fourier series form,applied to the first two terms,

$$g^{-1}(\theta_r, \phi) = \frac{1}{g_0\sqrt{1 - \delta_d^2)}} + 2\frac{1 - \sqrt{1 - \delta_d^2}}{g_0\delta_d^2\sqrt{1 - \delta_d^2}}cos(\phi - \theta_r) \tag{3.31}$$

Using these functions and equations, we can compute the self inductance and mutual inductance of the stator phase winding. Also if we assume the permanent magnet rotor to have a constant winding function throughout, we get the required inductance between the stator and rotor phase windings. The derivation of air gap for a system with static and dynamic eccentric can be found in this paper [29]. The system equations are as follows:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} r_s & 0 & 0 \\ 0 & r_s & 0 \\ 0 & 0 & r_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + w_r \begin{bmatrix} e_a(\theta_r) \\ e_b(\theta_r) \\ e_c(\theta_r) \end{bmatrix} + \begin{bmatrix} L(\theta_r) & M(\theta_r) & M(\theta_r) \\ M(\theta_r) & L(\theta_r) & M(\theta_r) \\ M(\theta_r) & M(\theta_r) & L(\theta_r) \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

It must be noted that calculation of inductance at each step is computationally cumbersome and must be performed before running the program. The values need to calculated with a very low time step to avoid any kind of biasing and to get accurate results. The stator-stator and stator-rotor self- and mutual inductances are calculated once and then used from a look-up table.

3.3.4 Frequency domain

Using the equations derived above, when the motor is run with and without the eccentricity fault, variations in the stator currents are recorded. The Fast Fourier Transform (FFT) and Continuous Wavelet Transform (CWT) results for the same are illustrated below in Figure 3.13.

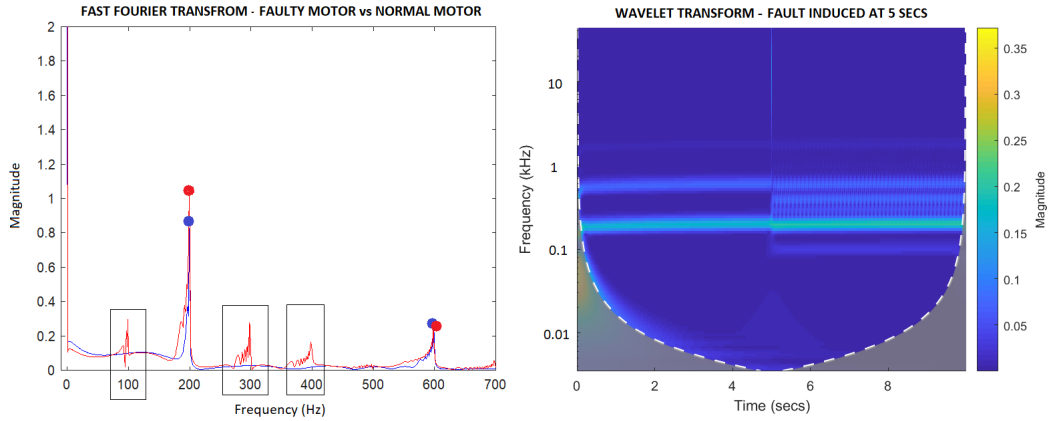


Figure 3.13: Frequency Signature of motor (blue) Healthy (red) Eccentric rotor

Motor current signature analysis (MCSA) has been the most popular approach considered for the detection of PMSM eccentricity faults, as it provides continuous online monitoring independent of motor parameters without additional sensors. In [6], The rotor rotational speed frequency component,is suggested as an indicator for detecting rotor eccentricity problems:

$$f_{ecc} = f_c\left(\frac{1 \pm k}{p}\right) \tag{3.32}$$

where f_{ecc} and f_c are the rotor eccentricity and fundamental frequency components, respectively, k is an integer, and p is the number of pole pairs.

3.3.5 Summary

In this chapter two faults have been modelled. The two faults being stator inter-turn fault and eccentricity fault. The theory behind the faulty models have been described and derived to make a clear transition from a normal to faulty situation. Frequency domain results of both the faults have been provided to look out for what to expect.

In the next chapter, the results of these faulty models are discussed. A brief introduction to wavelet transform (WT) is also provided. The frequency domain results obtained above were for stationary signals and we will be dealing with variable speed motors and hence need to adopt use of the WT.

Chapter 4

Simulation Results

4.1 Description of Computational Code

The programs were written using MATLAB (MathWorks, Inc., Natick, MA), and the full codes is provided in the appendix. There are two main programs, one for stator inter-turn fault and the other for rotor eccentricity fault. Each main program calls a function that contains the dynamic equations of that particular fault. The future states are evaluated using a RK4 solver that calls the respective function and gives it the required initial conditions.

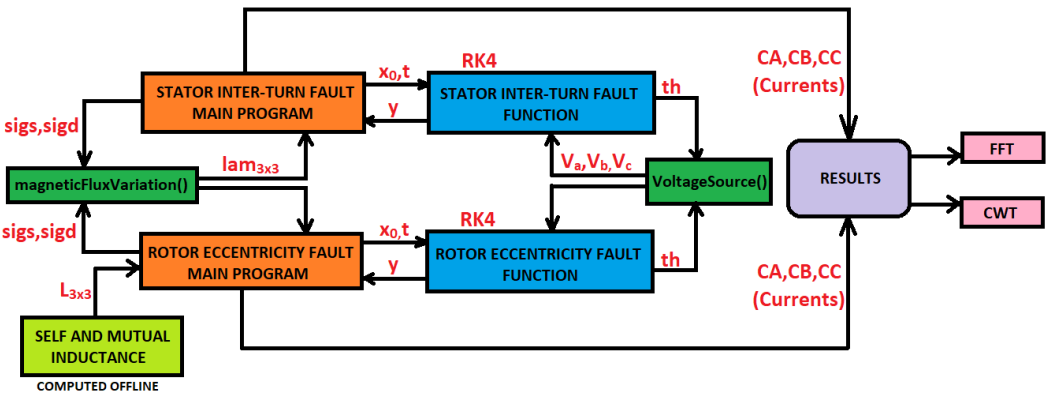


Figure 4.1: Flow of information between the different program segments

The main program also calls the magneticFluxVariation() function to determine the flux linkage for a system versus the rotor angle for given values of static and dynamic eccentricity (sigs,sigd). These values are a constant over rotor angle when there is no eccentricity present.

The respective functions for each fault also need to calculate the voltage supplied while evaluating the states. The VoltageSource() function takes the electrical angle as an input and gives the consequent voltage for phase A, B and C. The magnitude of the voltage is determined using a PD controller utilizing the difference in speed as the error function to be corrected.

For an eccentricity fault, apart from the permanent magnet rotor flux linkage, the self and mutual inductance value fluctuates as well. These values are already calculated for the different rotor angles using the 'self and mutual inductance' program. The values are used in the function from the lookup table generated offline.

The current values gathered from the simulation are sent to two programs: FFT and CWT. FFT gives the Fourier transform of the stator current signal sent to it, and the CWT performs a wavelet transform followed by the technique developed in this thesis. It provides a plot of the occurrence of frequency ratios versu frequency ratio. Also, it implements an algorithm to predict which fault is more dominant in the motor. The simulation results are attached in section 4.3.

Section 4.1, elaborately describes the important aspects of the code and also provides us with a table for motor parameters. Section 4.2 emphasizes on the need for wavelet transform and explains the theory behind it for an intuitive understanding. Also, it introduces the implementation of the main idea of this thesis. Section 4.3 presents the simulation results discussing the essential features to observe in these graphs. The chapter is designed to appreciate wavelet transform when dealing with variable speed and provide results to prove the working of the technique developed. The last section is added to encourage the idea of using neural

networks to improve the efficiency of the technique over time.

4.1.1 Motor parameter values

The brushless DC Motor modeled in MATLAB has a very simple configuration. A table of values is given below [4].

Physical parameter	Symbol	Value	Unit
Stator resistance	rs	2.75	ω
Leakage inductance	L_{ls}	1.1	mH
Magnetising inductance	L_{ms}	7.33	mH
PM Stator inductance	λ_m	8.62e-3	mH
Motor inertia	Jm	4e-5	kgm^2
Load inertia inductance	Jl	5e-5	kgm^2
Motor damping inductance	Bm	1e-6	$\frac{Ns}{m}$
Load damping inductance	Bl	1e-6	$\frac{Ns}{m}$
Number of Poles	P	2	-

Table 4.1: Motor Parameters

4.1.2 The main program and function

The model of a BLDC motor consists of two parts: The main program and the BLDC motor function. The main program declares and initializes the required motor parameters that remain a constant throughout the program, for example, resistance of the winding. It also assigns initial values to the states of the system. The main program calls the BLDC motor function to perform repetitive iterations over a fixed time step using an ode solver.The function formulates the states of the system. Finally, the states and interesting parameters of the system are displayed to it's understand.

The BLDC motor function is a vital part of the system. It contains all the system equations which have been derived in the earlier sections. The function is called with a given set of initial conditions passed by the main program. The required

motor speed is decided here and states are updated at each iteration accordingly. The function also calls the VoltageSource function, which provides trapezoidal voltage signals. Finally, important parameters are saved to be passed back to the main code for effective visualization of the system dynamics.

The control technique adopted for this motor is simple velocity control, using a proportional and integral controller. The respective Kp and Ki can be tuned in the main function to get the desired output. The output characteristics of the motor was made to follow a first order response (without overshooting), adjusting proportional gain to avoid overshoot. The value of Kp was set at 0.2 and Ki at 0.1. With these gains for the given motor parameters, it takes about four seconds for the motor to reach the required steady-state velocity.

Two separate programs have been developed, one for each fault, as both the programs have different number of states. Nevertheless, they are very similar with just a few differences. These faulty models were derived in the previous chapter.

4.1.3 Healthy motor: States, inputs and outputs

The block diagram given below contains all the steps in the program. It offers a visual aid to interpret the direction of information flow. This provides a better sense of the whole process, as it is encapsulated in one diagram.

The states of the system are the three flux linkages (one from each phase) $\lambda_a, \lambda_b, \lambda_c$, the angular velocity of the rotor ω_m, the rotor angular position θ_r, the electrical angular position θ_e and error sum ie. We can go back to the block diagram given above and look at the states we mentioned above following the integrator blocks. Another state we need to keep a track of is the summation of error state ie, in order to implement integral control. The outputs that are logged at the end of each iteration are: the three currents i_a, i_b, i_c, the torque acting on the rotor T, the angular velocity of the rotor in revolutions per second $\frac{\omega}{2\pi}$, error percentage, the

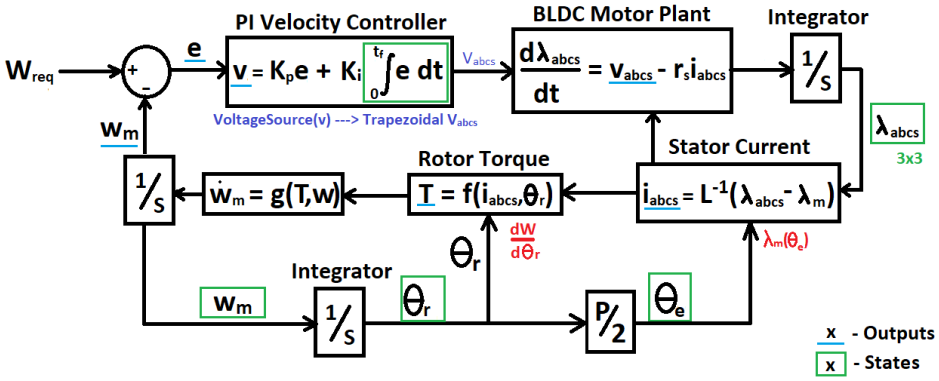


Figure 4.2: Block Diagram of the Brushless DC Motor function

control input to the circuit (maximum voltage magnitude) and the three voltages V_a, V_b, V_c.

The only input provided to the system is the required angular speed ω_{req}. If we look at the system in more detail we realize that there are a number of parameters we need to assign a value to, eg. the winding resistance, inductance matrices, controller constants etc. These values are provided in section 4.1.

4.1.4 Trapezoidal voltage source

The 'VoltageSource()' is a function called in ever iteration to update the voltage value in each phase. The control output is responsible to look at the speed error still present and output a voltage to as to minimize the error. As we are dealing with a 3 phase system we need three voltages that are 120 degrees apart, for the motor to run normally.

The VoltageSource() however just takes into account the electric angle. This angle in reality is a feedback from the hall effect sensors that are monitoring the position of the permanent magnet. The angle is calculated to be a number between 0 and 2π and passed to the Voltage Source as an input. This is done by using the

Theta_elec (θ_r)	x	y	z	$f_a(\theta_r)$	$f_b(\theta_r)$	$f_c(\theta_r)$
0^0 - 60^0	1	0	1	1	-1	$1-\frac{6\theta_r}{\pi}$
60^0 - 120^0	1	0	0	1	$\frac{6\theta_r}{\pi}-3$	-1
120^0 - 180^0	1	1	0	$5-\frac{6\theta_r}{\pi}$	1	-1
180^0 - 240^0	0	1	0	-1	1	$\frac{6\theta_r}{\pi}-7$
240^0 - 300^0	0	1	1	-1	$9-\frac{6\theta_r}{\pi}$	1
300^0 - 360^0	0	0	1	$\frac{6\theta_r}{\pi}-11$	-1	1

Table 4.2: Table with hall sensor readings and Voltage readings for the corresponding angles

$rem()$ function. Surely there may be more ways to do this. The voltages v1, v2, and v3 are selected according to angle and its maximum magnitude in 1. These three voltage values are in the form of trapezoidal waves. The three voltages are multiplied to the controller output to get the actual voltage in the three phases. Table 4.2 tabulates how the voltage changes with the hall effect sensor readings.

4.1.5 Stator inter-turn fault program

Difference in main program

1) Two new variables are needed to induce a SITF fault in the system, i.e. mu and Rf. mu is used to denote the fraction of the wire that is shorted and Rf is the faulty resistance attached to the shorted path. If the value of Rf is small, most of the current flows through it. This induces an opposite flow of current in the shorted region, also wasting the potential of 'mu' times the winding. However if 'Rf' is huge, i.e., it is a high resistance path, the fault is not very serious.

2) Also the matrix L is now $4x4$ (Refer back to 3.2.3). The resistance matrix is also $4x4$. One extra state is added to the system which is the flux linkage in the fault winding loop which needs to be initialized to zero as well.

3) In the plots to be displayed, there is one extra current output collected i.e. i_f. This is the current through the shorted path. Lesser the i_f, less severe the fault. The Rf is chosen to be 1 ohm and mu is varied from 0.6-0.9.

Difference in function

1) The function contians an extra state which is the flux linkage in the shorted loop lf. The stator rotor inductance (back emf) will also undergo a change in dimension (become 4X1 rather than 3x1) as it now contains a new term accounting for the shorter winding inductance (proportional to mu).

2) An extra variable is logged as an output i.e faulty current if. The voltage matrix which also needs to a 4x1 to stay consistent with dimensions, gets its values from the VoltageSource() function and uses the fourth value zero. This is because potential difference across a point on a loop is zero (Kirchoff's law).

3) Not to forget the extra current also finds its way into the torque equation as it does takes up a part of the energy.

Note: Small values of mu and Rf, yields large L, and are hence avoided.

4.1.6 Eccentricity fault program

Difference in main program

1) Two new terms are used in the main program which are *sigs* and *sigd*. These two quantities help us determine the amount of static and dynamic eccentricity present in the model, respectively. Both are a number varying from 0 to 1, where 0 denotes a healthy motor and 1 denotes entirely faulty.

2) The stator-stator inductances are derived in any entirely different function for a static eccentricity of 0.6 and dynamic eccentricity of 0.2. Approximate but very close values are used here (approximated to a sine wave). These values are a function of rotor position.

3) The stator-rotor inductances are derived from a function as it is calculated fast. We shall elaborate on this function $magneticFluxVariation()$ later, in greater detail. The function accepts *sigs* and *sigd* as input arguments.

Difference in function

1) As the stator-stator and stator-rotor inductances vary with rotor angle, we need to use the rotor angle to pick the right inductances from the table of calculated inductances. So we use a variable to save this angle that lies within the set [0,2*pi).
2) As mentioned above, stator-stator inductances are approximated as sine waves in the presence of an eccentricity. When there is no eccentricity the inductance stays a constant.
3) The stator-rotor inductances are derived in the main program and passed as a global variable, to be used within the function.
4) Different results can be obtained by varying the variables *sigs* and *sigd* to witness the difference between the three conditions. These values helps implement static eccentricity, dynamic eccentricity and mixed eccentricity.

magneticFluxVariation()

This function is used to calculate the stator-rotor inductances. It takes the static eccentricity (e_s) and dynamic eccentricity (e_d) as an input. These two parameters change the air gap function. The air gap function is rotor angle dependent when dynamic eccentricity of any amount is present. The inverse of air gap function is given by the equation below,

$$g^{-1}(\phi, \theta_r) = G_0 + G_1 cos(\phi - \alpha) + G_2 cos(2(\phi - \alpha)) \tag{4.1}$$

$$where, \quad G_0 = \frac{1}{g_0\sqrt{1-e^2}}; \qquad G_1 = \frac{2}{g_0\sqrt{1-e^2}}\left(\frac{1-\sqrt{1-e^2}}{e}\right)$$

$$G_2 = \frac{2}{g_0\sqrt{1-e^2}}\left(\frac{1-\sqrt{1-e^2}}{e}\right)^2; \qquad \alpha = tan^{-1}\left(\frac{e_d sin\theta_r}{e_s + e_d cos\theta_r}\right)$$

$$e = \sqrt{e_s^2 + e_d^2 + 2e_S e_d cos\theta_r}$$

Here, where ϕ is the stator circumferential angle, θ_r the rotor angle in mechanical degree and g_0 the length of the air-gap in symmetrical condition.

This function assumes the mmf due to the permanent magnet to be a constant which is similar to saying that the number of turns involved in the linkage stays a constant throughout. It just oscillates from positive to negative while going from north to south pole. It calculates the inductance using the idea in [23].

4.2 Fault detection algorithm

4.2.1 Need for variable speed in the industry

1) Since motors consume a majority of the energy required to operate, the control of motors, based on demands of loads, increases in importance, as energy supplies become ever more strained. Additionally, end users of motors can realize $25 - 70\%$ energy savings by using motor controllers.

2) A few advantages of using an AC motor drive is that when the motor starts the current demanded is high. This is as much as seven-to-eight times the motor full-load current. This also places an enormous drain on the power distribution system connected to the motor. Typically, the supply voltage sags, with the amplitude of the sag being dependent on the size of the motor and the capacity of the distribution system. These voltage sags can cause sensitive equipment connected on the same distribution system to trip offline due to the low voltage.

3) An AC motor started across the line is a tremendous mechanical shock both for the motor and connected load. This shock will, over time, increase the wear and tear on the connected load, as well as the AC motor.

4) A variable speed driver is also preferred because it allows adjusting the operating speed, set a torque limit and implement controlled stopping. These features are essential for flexible control, safety of components and reducing wear and tear.
5) It helps in saving energy. Centrifugal fan and pump used to cool follow a variable torque load profile, which has horsepower proportional to the cube of speed and torque varying proportional to the square of speed. As such, if the speed of the fan is cut in half, the horsepower needed to run the fan at load is cut by a factor of eight $(1/2)3 = 1/8$. At constant speed the motor would still be running at full load and full speed (full power). Energy savings with a variable speed motor can be sufficient to pay back the capitalized cost in a matter of a couple of years.
6) Using an Adjustable Speed AC Drive eliminates the need for a reversing starter, since the output phases to the motor can be electronically changed without any mechanical devices. It also eliminates the need for mechanical transformers eg. gearboxes. Because the AC Drive can operate with an infinite variable speed. It can deliver the low- or high-speed required by the load. This eliminates maintenance costs, as well as reducing floor-space requirements.

4.2.2 Signal requirements to perform Fast Fourier Transform

FFTs gives us the frequency information of a signal, which means that it tells us how much of each frequency exists. But, it does not tell us when in time do these frequency components occur. This information is not required when the signal is 'stationary'.

Let's take a closer look at the concept of stationarity, since it is of paramount importance in signal analysis. Signals whose frequency content does not change in time are called stationary signals. In this case, one does not need to know the time at which a frequency component arises, since all frequency components exist at all times.

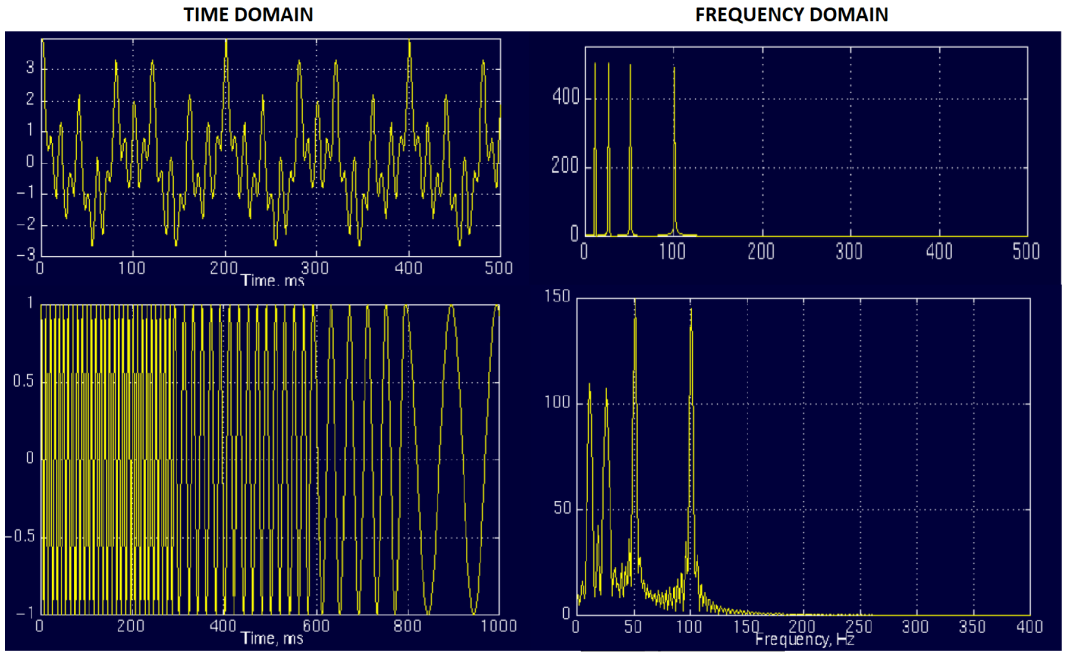


Figure 4.3: Failure of FFT to differentiate between the two time signals[14]

The two examples below would help explain this better. Both the time signals give more or less the same result even though they are quite different. Hence we see that all perspective of time is lost in the frequency domain and we need to apply the condition of stationarity.

4.2.3 Need for Wavelet transform

It is clear that FFT are good transforms to map a time signal into the frequency domain only if the signal is a stationary signal. When stationarity is not guaranteed, wavelet transforms prove to be more useful.

If we don't have a stationary signal, we will get an FFT that depends on which portion of the signal being analyzed. Imagine the motor current of a robotics motor, running at different speeds from time to time. The FFT of this current signal will have a peak at all the speeds the motor ran at. This will be similar to a motor

that has all these frequencies due to a fault but running at a constant speed. This is how the FFT fails to be useful in our case. We don't have a unique time signal to map back to if the time signal is not considered stationary.

Now, the first idea that comes to someone's mind when this problem is discussed is why don't we perform multiple FFTs for small time periods. This is a logical thought, and this approach has been used actively (STFT) for a while until wavelet transforms were discovered. It is understandable to conclude that the shorter the time set, the better will be the results, as we don't know at the frequency the results are changing to and from. So the higher the accuracy, the better it this method works. But there is a drawback. We can never get a full idea of frequency and time. If we want a high resolution in frequency the resolution in time is compromised and vice versa. Hence, we need to decide on an optimized time interval to get a good resolution in time and frequency. STFTs use square windows of time and frequency. Wavelet transform is an upgrade to this approach. Wavelet transform gives you good resolution in time for high frequencies and good resolution of frequencies for low frequencies which proves better than a square window size in STFT.

4.2.4 Theory for wavelet transform

A brief theory of wavelet transform can be found in the MATLAB documentation or in various online sources. A part of this theory relevant to be discussion is included here from [35]. The mathematical expression used to perform Fourier transform is as given below:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \tag{4.2}$$

which is the integral of the product of the time of the signal f(t) and a complex exponential (a complex exponential can be broken down into real and imaginary sinusoidal components which can be visualized as points on a circle.).

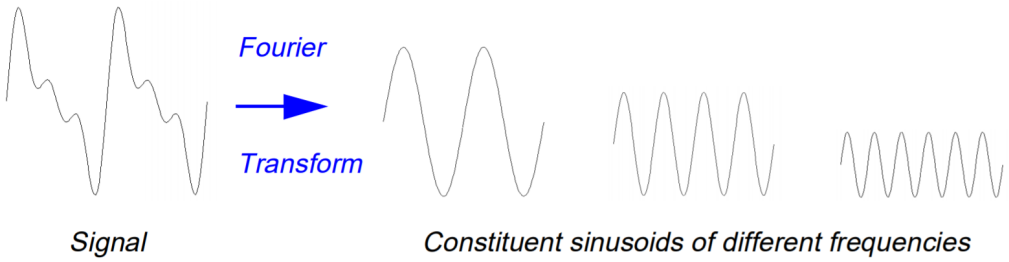


Figure 4.4: Graphical representation of Fourier Transform[35]

As we know that every wave can be broken down into multiple sine waves. Fourier transform gives us a coefficient corresponding to each frequency, that represent the amount of a particular frequency present in the wave. Graphically, the process appears as in Figure 4.4 Similarly, the continuous wavelet transform (CWT) is defined as the integral over time of the product of the time of the signal and shifted and scaled wavelet function: The result of the CWT are many wavelet coefficients C, which are a function of scale and position.

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\Phi(scale, position, t)dt \tag{4.3}$$

Graphically CWT looks like this:

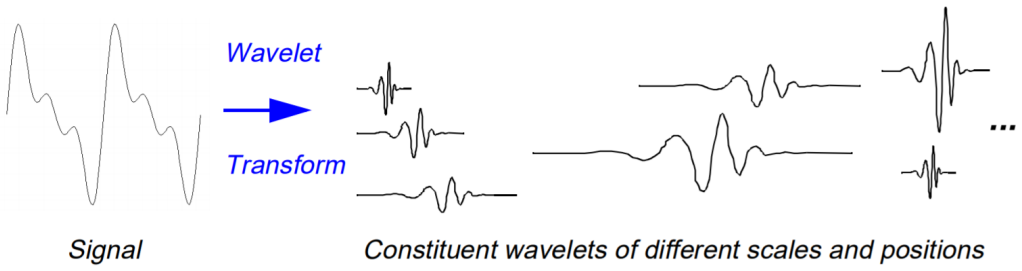
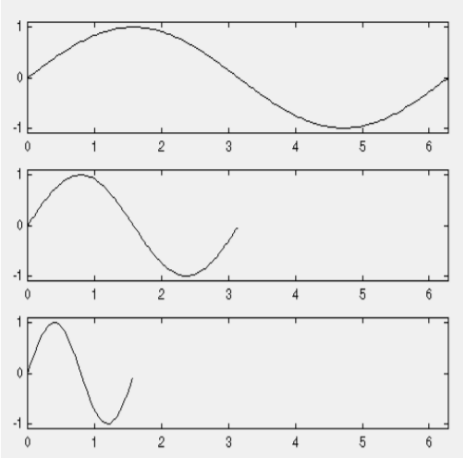


Figure 4.5: Graphical representation of Continuous Wavelet Transform[35]

Now let's discuss scaling and shifting. Scaling a wavelet simply means stretching or compressing it. Scaling is carried out by the scaling factor 'a' (used here). Scaling basically increases and decreases the frequency of the mother wavelet. Let's look at

the affect of scaling on sinusoidal waves for clarity:



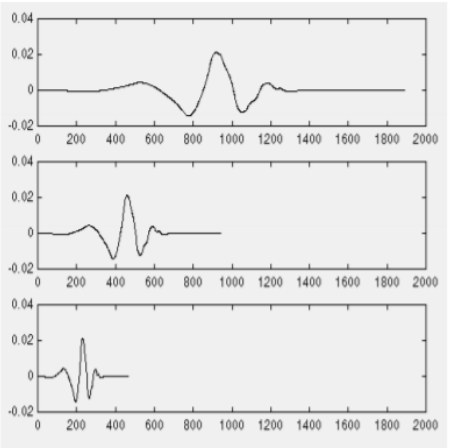
$$f(t) = \sin(t) \quad ; \quad a = 1$$

$$f(t) = \sin(2t) \quad ; \quad a = \frac{1}{2}$$

$$f(t) = \sin(4t) \quad ; \quad a = \frac{1}{4}$$

Figure 4.6: Scaling a sine wave[35]

The scale factor works exactly the same with wavelets. The smaller the scale factor, the more "compressed" the wavelet, but thinking about it in the frequency domain. The smaller the scaling factor, the higher the frequency. The figures should help make sense of the effects of scaling on the frequency of the wavelet.



$$f(t) = \psi(t) \quad ; \quad a = 1$$

$$f(t) = \psi(2t) \quad ; \quad a = \frac{1}{2}$$

$$f(t) = \psi(4t) \quad ; \quad a = \frac{1}{4}$$

Figure 4.7: Scaling a wavelet[35]

Shifting, on the other hand, just means means delaying (or hastening) its onset. It is necessary for the wavelet to span the entire course of time of the signal.

Graphically, the delaying of a wavelet by k can be represented by:

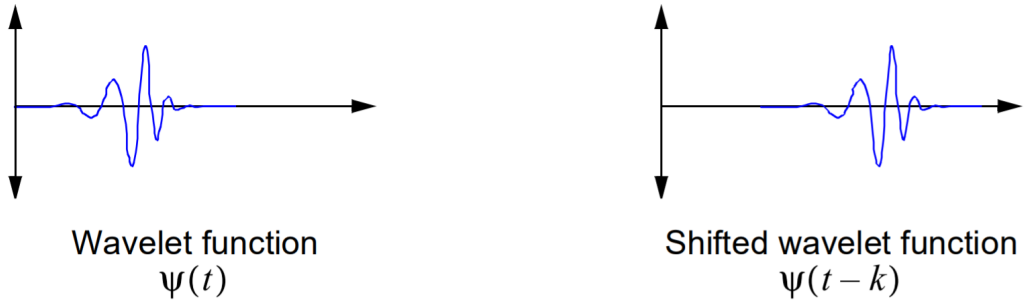


Figure 4.8: Shifting a wavelet[35]

4.2.5 Description of the method developed

An enormous amount of work has been carried and reported on in the literature regarding fault detection and diagnosis of a motor. A majority of papers involve experimental verification of faults through the presence of an extra frequency in the spectral distribution. These papers provide insight into what to look for, namely patterns in the abnormal frequencies that stand out. There is also a lot of supporting theory behind the reason for that frequency to be present. One thing to note about these frequencies is that they are always a function of the central frequency. Further, the central frequency is a function of the motor speed and pole pairs, typically the peak with the highest amplitude.

A majority of the results reported are from experiments carried out at constant speeds. The amount of material present on variable speed drivers is minimal due to the relatively recent advent of wavelet transform. Using wavelet transform gives the capability of deriving an algorithm that can pick up faulty frequencies even at variable speeds.

This algorithm works on the idea of frequency ratios and not frequencies.

It assigns the value of unity to the central frequency and takes a ratio of all the frequencies with respect to it. By adopted this way, no matter what speed the motor is working at, the central frequency will always be at 1. When a fault occurs, there are two possibilities. One, the peaks of some of the frequency ratios grow higher or two, a new frequency ratio comes up. Now, since we need to be compatible in working with real world data, and real data has noise we can't consider all the peaks that come up. So the algorithm is designed to record the 6 highest peaks of frequencies at every time interval. It then assigns 1 to the central frequencies and finds the 5 frequency ratios.

It is also suggested to decide a threshold below which no peaks are considered to avoid noise being picked up as peaks when the motor stops. At the end it plots the occurrence of a frequency ratio versus frequency ratio. It is observed that in the presence of an eccentricity fault we get a few new frequency ratios (even multiples and plus or minus half the central frequency) whereas in the case of a stator inter-turn winding fault we notice a rise in the already present harmonic frequencies i.e. old multiples of the central frequency.

4.3 Results

The main program of the healthy BLDC motor, as derived above, is designed to collect essential information from the function after each iteration. It helps us comprehend the working of a Brushless DC motor by capturing the trends of the different parameters. Deriving these characteristics also helps verify the results, with well known literature on this topic[1]

4.3.1 BLDC: Angular velocity, Control Output & Error Percentage

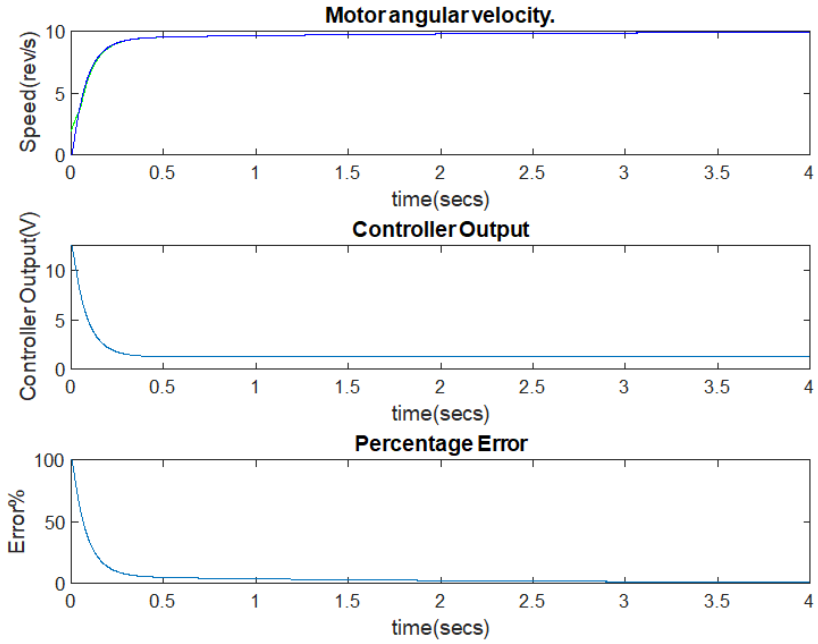


Figure 4.9: Output of a healthy motor:(top to bottom)Angular velocity of the rotor; Controller output that becomes applied voltage magnitude; and Angular velocity error percentage

The motor starts from rest and accelerates up to the required speed of 10 revolutions per seconds in about 4 secs. The response is tuned to look like a first order system.

The value of the proportional control constant K_p and integral control con-

stant K_i are set to a low value to achieve lower power requirements and lesser fluctuations. The integral is set to value almost an order of magnitude lower than the proportional control to avoid the systems from oscillating around the steady state value. The magnitude of maximum voltage supplied is equal to the controller output.

The percentage error is calculated as the fraction of speed difference (required - actual) and required speed, into 100. We notice it falling from 100 to 0 which makes proper physical sense.

4.3.2 BLDC: Stator Current, Phase Voltage & Rotor Torque

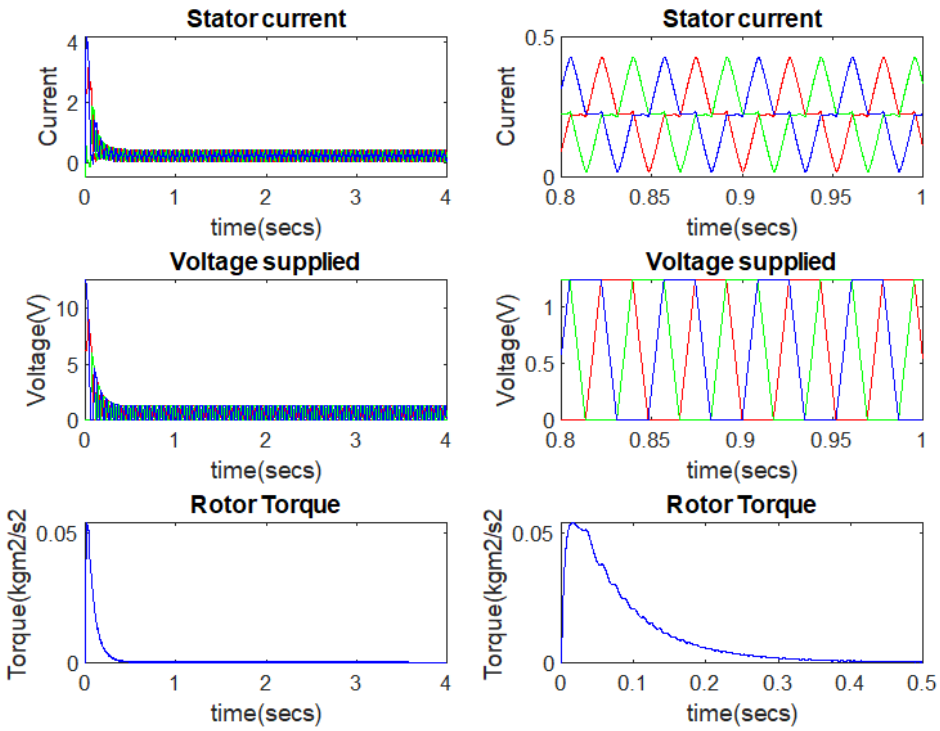


Figure 4.10: Output of a healthy motor: (top to bottom) Stator current in phase A, B and C; Voltage applied to phase A, B and C; and Torque experienced by the rotor (left to right) Full time result; left image magnified to observe essential characteristics.

As the motor starts from rest, the voltage supplied to the three phases are large

in the starting when the error is the highest. As the motor begins to achieve the speed, the voltage begins to reduce. Also, in the zoomed in version we notice that the voltage applied is trapezoidal in nature.

The current in each of the three phases follow the same trend. When the voltage is high so is the current. We notice that the voltage and current never go to zero. That would happen if there was no damping present in the system. In that case, once the speed was achieved no more input would be required. So the steady state of voltage and current occurs once the required speed is achieved.

Intuitive to understand, the torque acting on the rotor is really high in the starting when the rotor isn't moving. But once it begins to rotate the torque applied automatically reduces, falling to just the value required to resist the damping. From the figure, we can say that 0.01 secs is approximately the time when the rotor started rotating as that's where the torque value began to drop.

4.3.3 Step and sine velocity inputs

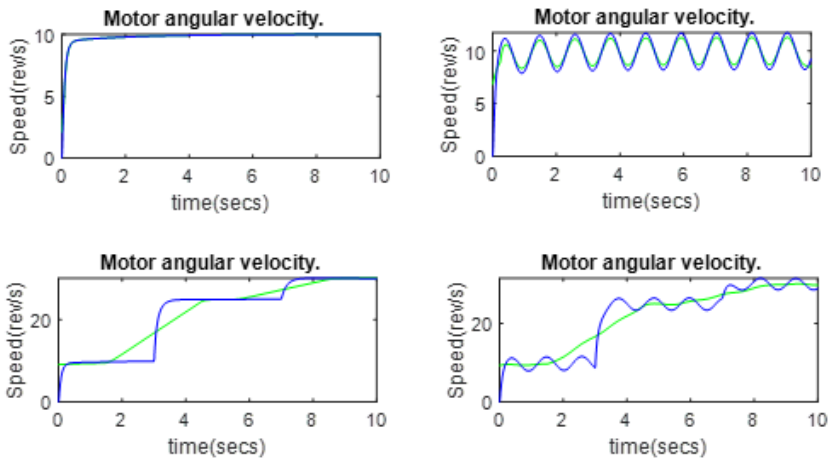


Figure 4.11: (Left to Right) Constant input; Constant plus sinusoidal input (Top to Bottom) One speed; Three speeds

Since we aim to detect and dignosis faults in the motor at variable speeds, we used

the above given speeds. At variable speeds the signals collected are no longer stationary. At variable speeds, a simple FFT will fail to pick up unique distinguishable characteristics to affirm the presence of a fault. To appreciate the method developed in the thesis, variable speeds are adopted.

The first case is a constant required speed of 10rev/s. The second case is a superposition of a constant speed of 10 rev/s and sinusoidal wave of amplitude 4 rev/s. In the third case, the initial required velocity of 10 rev/s is stepped to 25 rev/s at 3 secs and 30 rev/s at 7 secs. The fourth case is case three, superimposed with a sine wave of amplitude 4 rev/s.

4.3.4 Stator current in each phase with and without SITF

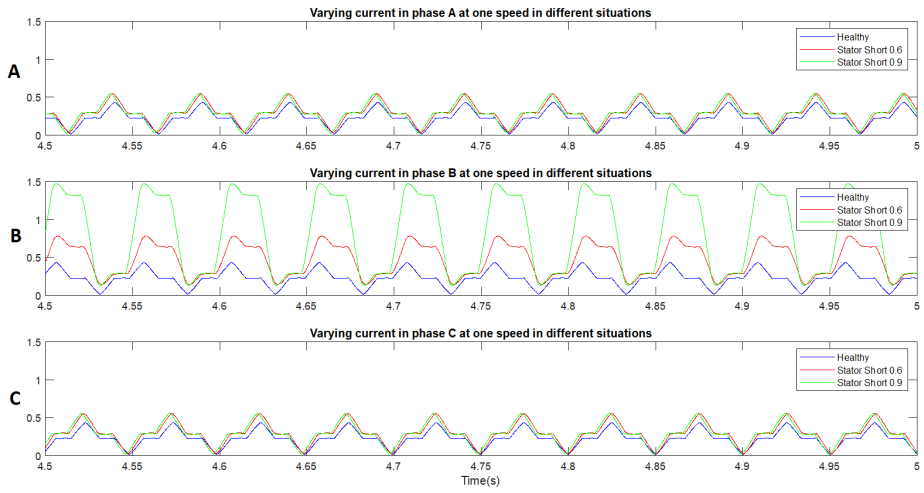
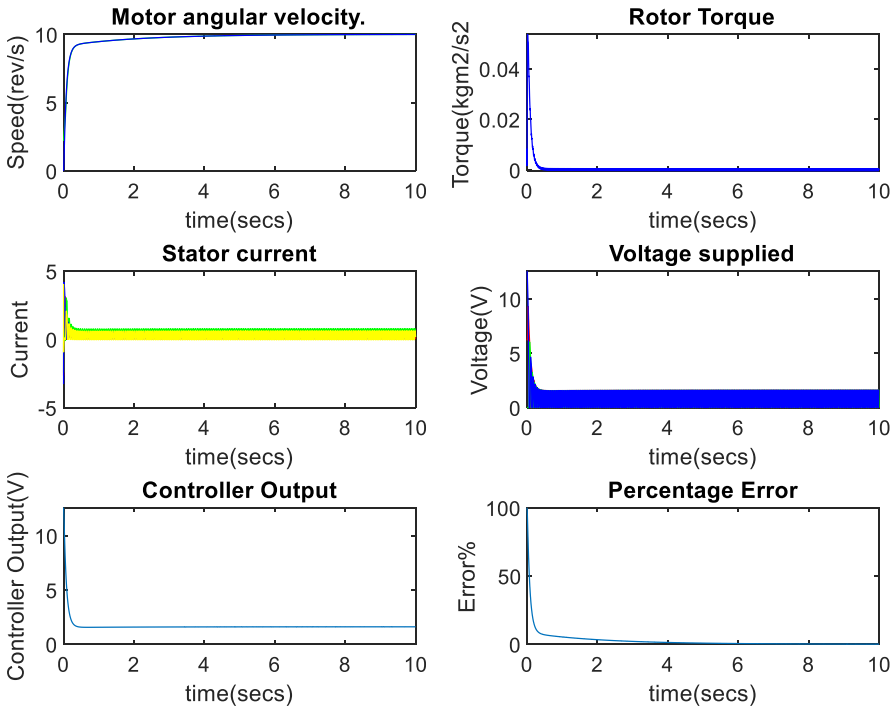


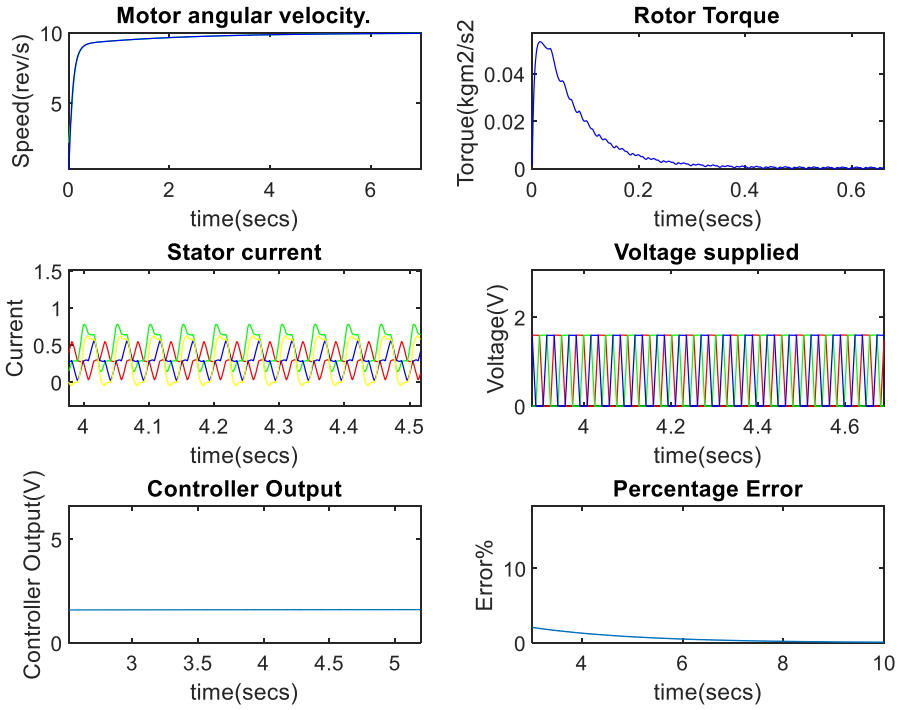
Figure 4.12: Phase currents through stator windings (top to bottom) A, B and C. (depicted in different colours) healthy motor, 60% of the stator winding shorted motor and 90% of the stator winding shorted motor

Current in A and C doesn't change much from the 0.6 to 0.9 case unlike in winding B where a drastic change is noted. It is obvious that more current is needed to create the same torque (as when healthy) when 0.6 to 0.9 the entire length of the wire is shorted (wasted). A and C depict a slight increase too.

StaShtOneSpeed



Zoomed in version



4.3.5 Stator current in each phase with eccentricity fault

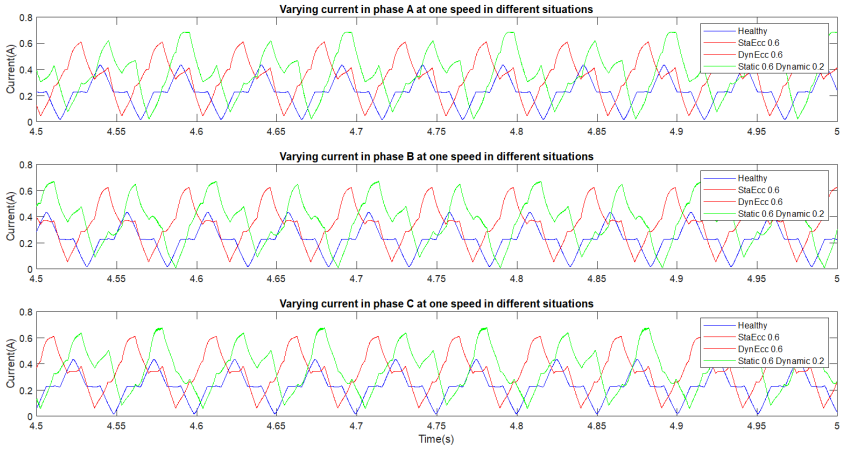
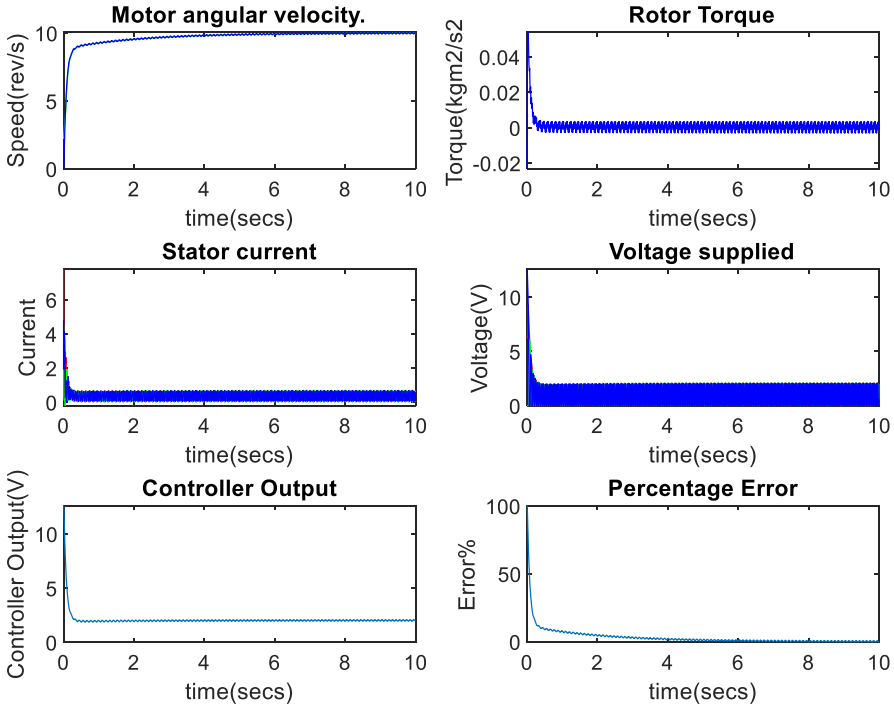


Figure 4.13: Phase currents through stator windings (top to bottom) A, B and C. (depicted in different colours healthy motor, 60% static eccentric motor, 60% dynamic eccentric motor and 60% static eccentric plus 20% dynamic eccentric motor

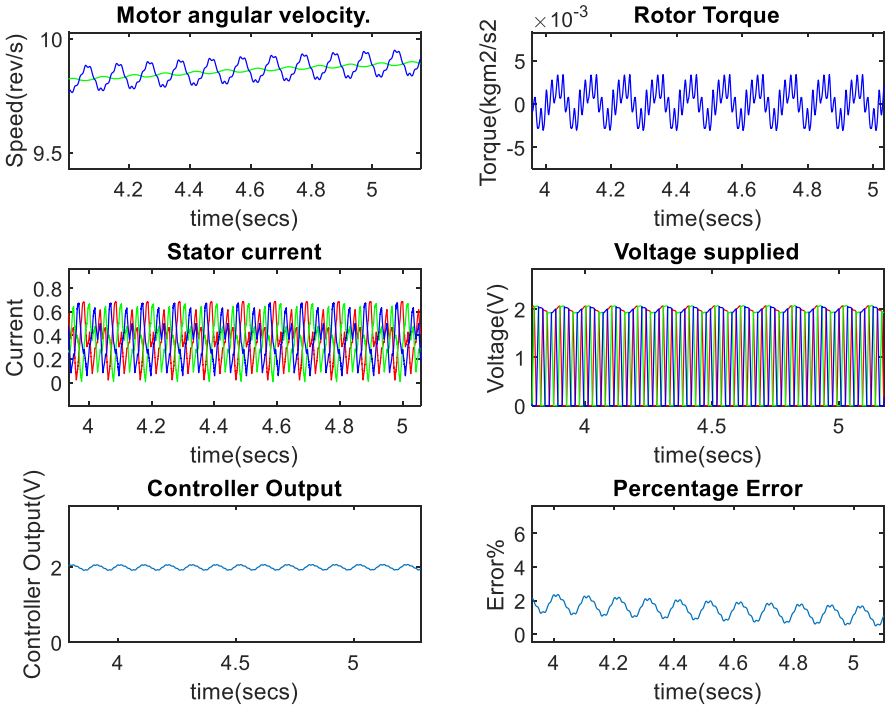
Unlike the previous case, the effect of the fault comes up equally in all the phases. We also notice that dynamic and static eccentricity current results are almost on top of each other. The green line is plotted for the case where there is mixed eccentricity. 20% dynamic eccentricity and 60% static eccentricity. If you look at that result closely, you will notice its amplitude increasing and decreasing periodically.

In order to implement static and dynamic eccentricity, two functions are implemented to calculate the different inductances associated with the system. One calculates the stator-rotor inductance and the other calculates the self and mutual inductances of the stator windings. Since the second function is computationally cumbersome the value is calculated and saved for use like a lookup table. The stator-rotor inductances are plotted below for different senarios and one plot of self and mutual inductances is calculated specification for the 20% dynamic eccentricity and 20% static eccentricity fault. Given is the output for the above mentioned senario. The zoomed in version depicts the oscillatory behavior of the current signal.

De2Se6OneSpd



Zoomed in version



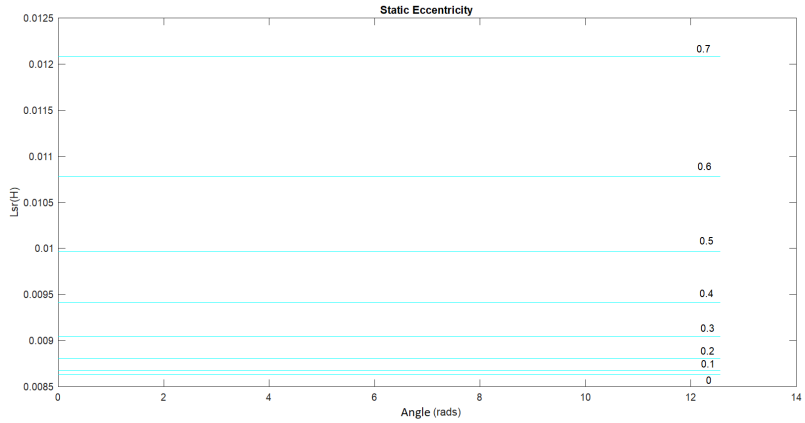


Figure 4.14: Stator-Rotor Inductance for varied static eccentricities

The figure above shows how the value of Lsr, i.e. stator-rotor inductance[20] [23] increases with eccentricity. The plots for static eccentricity of 0 to 0.7 are plotted above. It is seen that the plots for dynamic eccentricty of the same amount has the same plot. But zooming into the 0.7 eccentricity for static and dynamic eccentricity spells out the difference. The reason we only see blue lines i.e. Lsr(c) is because these lines Lsr(a), Lsr(b) and Lsr(c) are very close. The next figure depicts this closeness and also the diffeenced between the Lsr(a), Lsr(b) and Lsr(c) in the case of static and dynammic eccentricity.

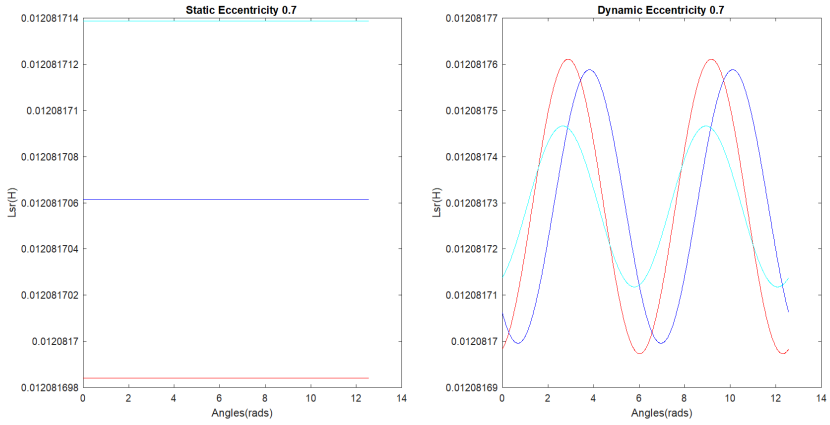


Figure 4.15: (Left to right) Static Eccentricity 0.7 , Dynamic eccentricity 0.7[33]

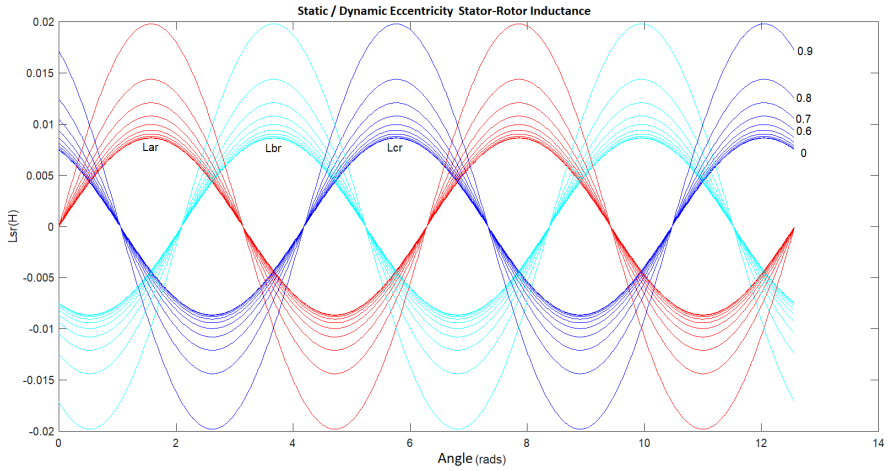


Figure 4.16: Flux Linkage between the permanent magnet rotor and stator winding A, B and C for different values of individual eccentricities (either static or dynamic)

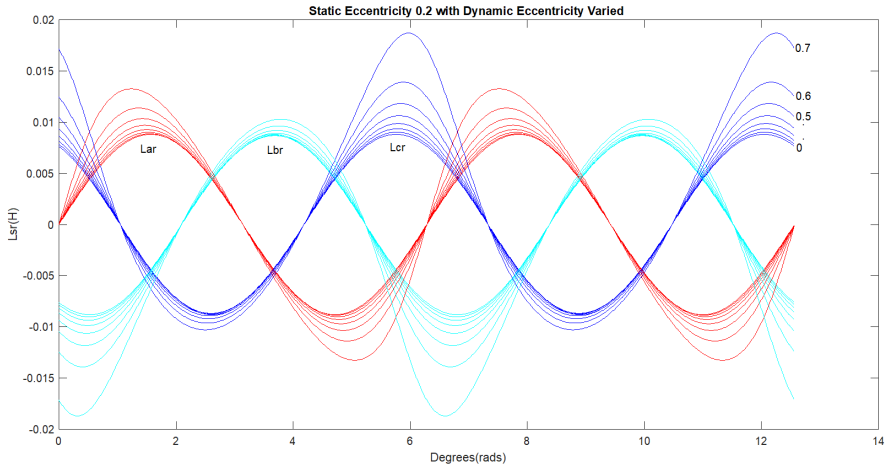


Figure 4.17: Flux Linkage between the permanent magnet rotor and stator winding A, B and C for static eccentricity of 0.2 and dynamic eccentricity varied from 0 to 0.7

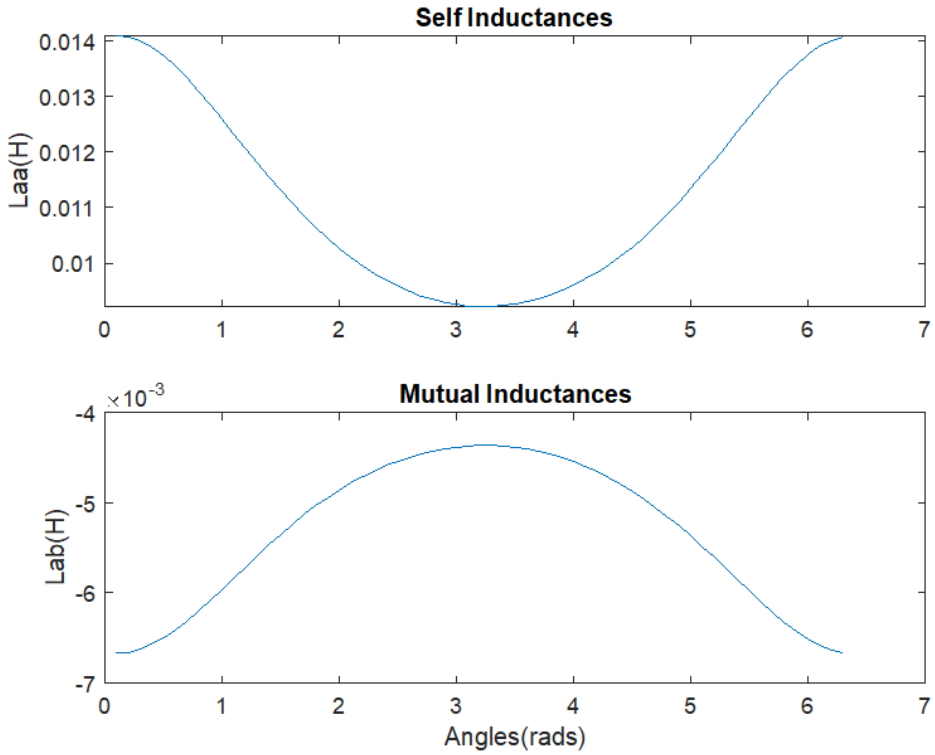


Figure 4.18: Self inductance L_{aa} and Mutual Inductance L_{ab} with SE 0.6 and DE 0.2

The inductance L_{ss} is equivalent to the value of the matrix below, so these two values are sufficient.

$$Lss = \begin{bmatrix} Laa & Lab & Lab \\ Lab & Laa & Lab \\ Lab & Lab & Laa \end{bmatrix} \tag{4.4}$$

4.4 FFT and Wavelet Transform Results

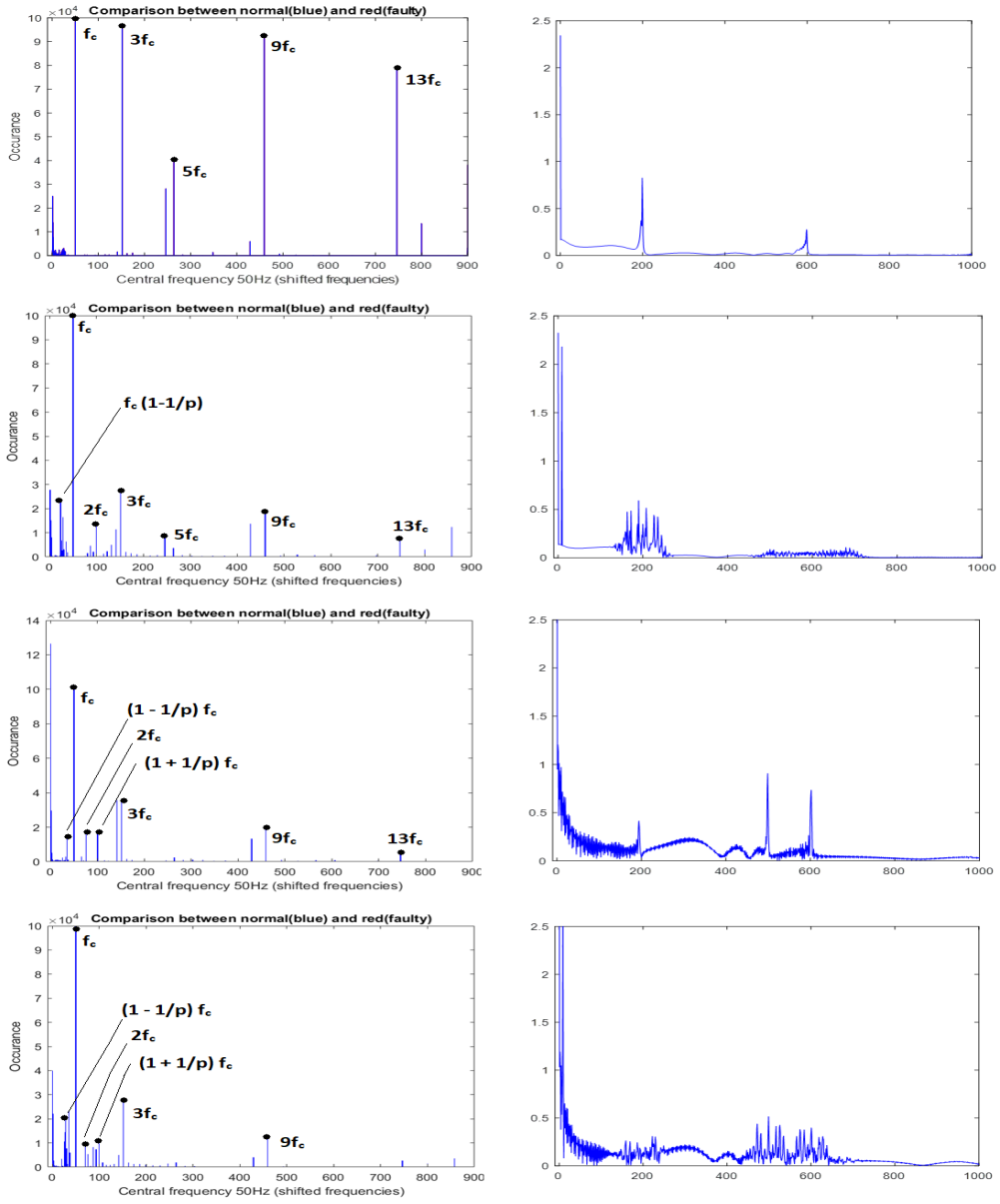


Figure 4.19: Healthy Motor
(Left to right)Prediction of which fault is present, Occurrence histogram of frequency ratios obtained from the Wavelet Transform, Fast Fourier Transform
(Top to Bottom) One Speed, One speed (sinusoidal), Three Speed, Three speed (sinusoidal)

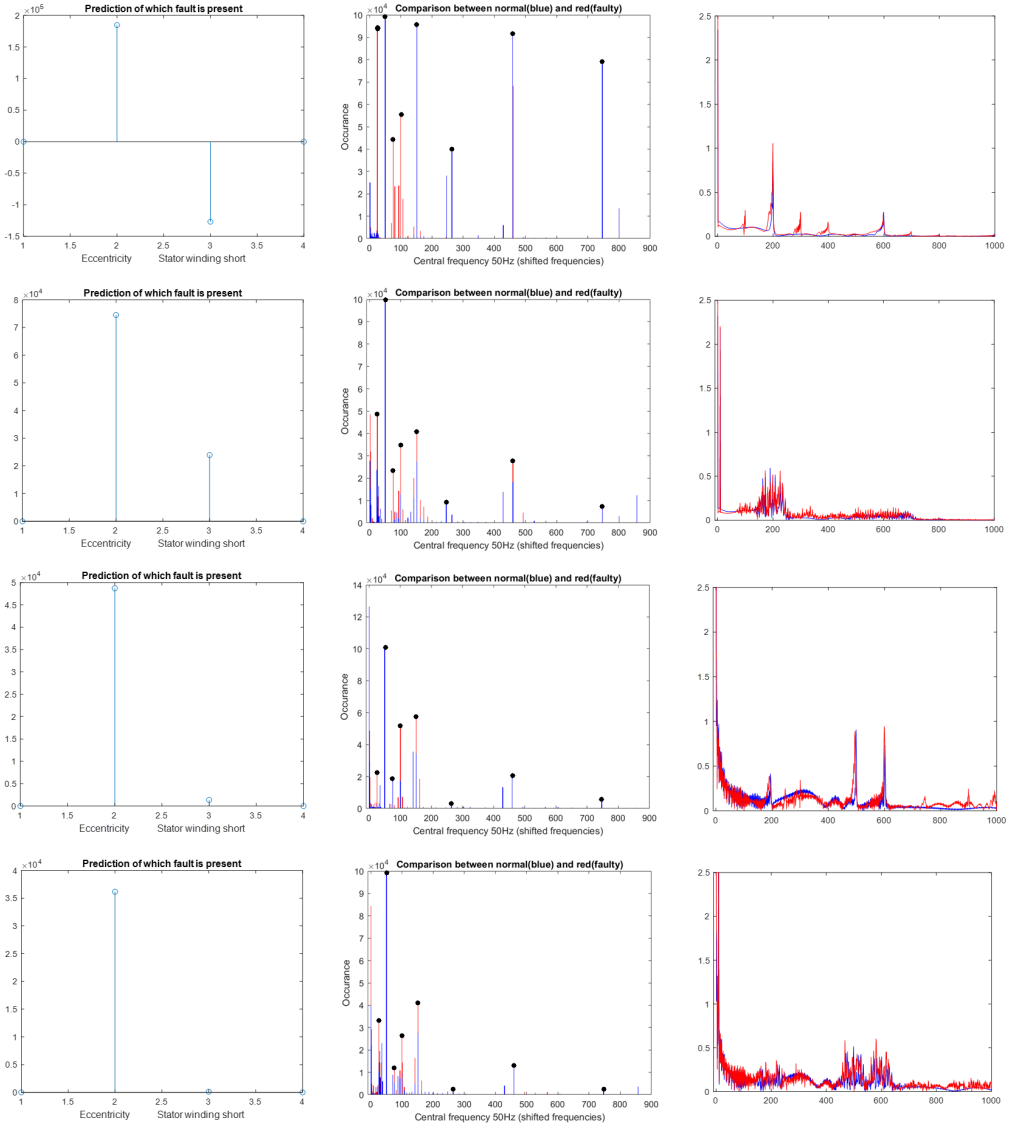


Figure 4.20: SE 0.6 DE 0.2, Notice $0.5f_s, 1.5f_s$&$2f_s$
(Left to right) Prediction of which fault is present, Occurrence histogram of frequency ratios obtained from the Wavelet Transform, Fast Fourier Transform
(Top to Bottom) One Speed, One speed (sinusoidal), Three Speed, Three speed (sinusoidal)

In the eccentricity faults, the values of frequency to look out for are given by,

$$f = f_c(\frac{1+k}{p}) \qquad \text{where k=1,2..p and p=pole pairs}$$

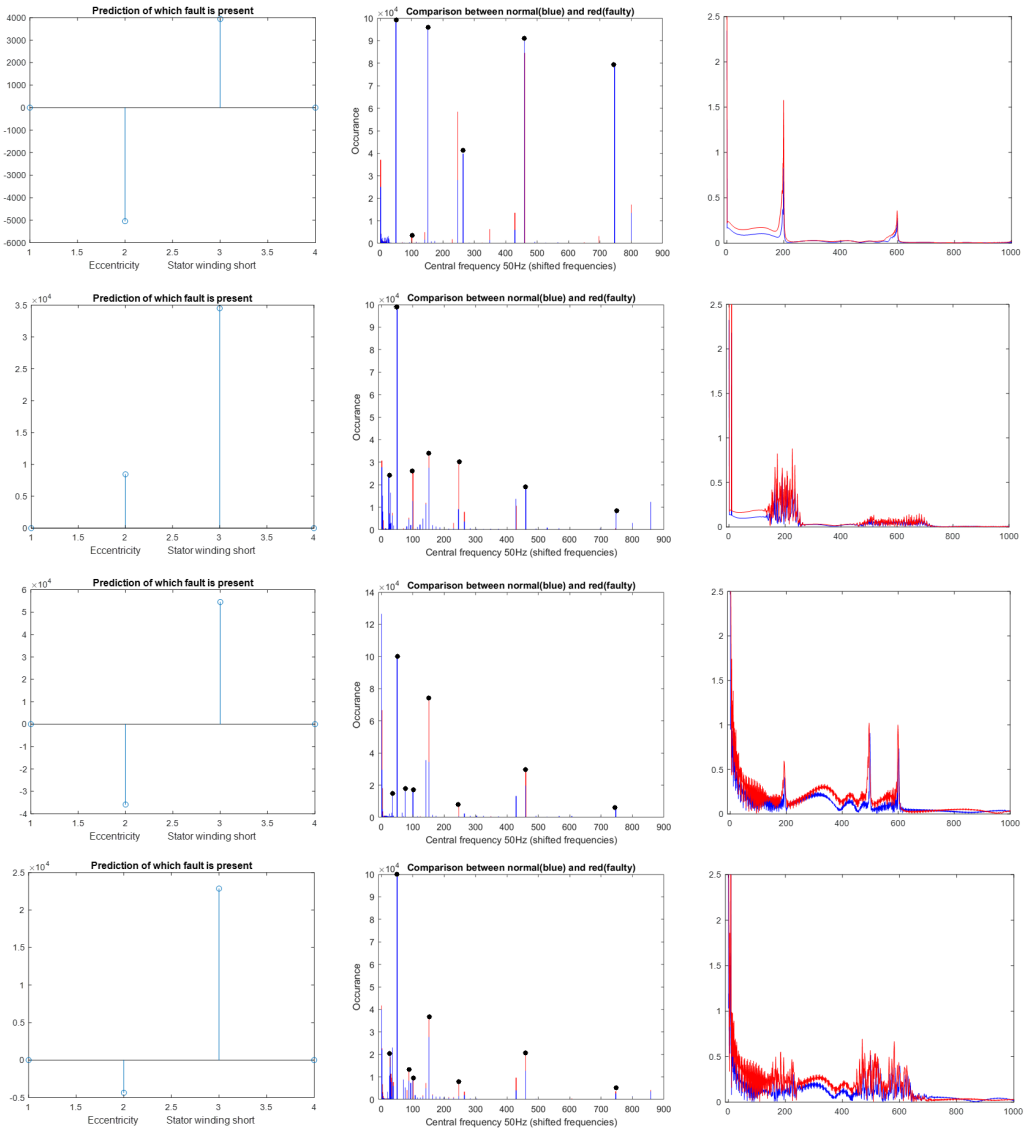
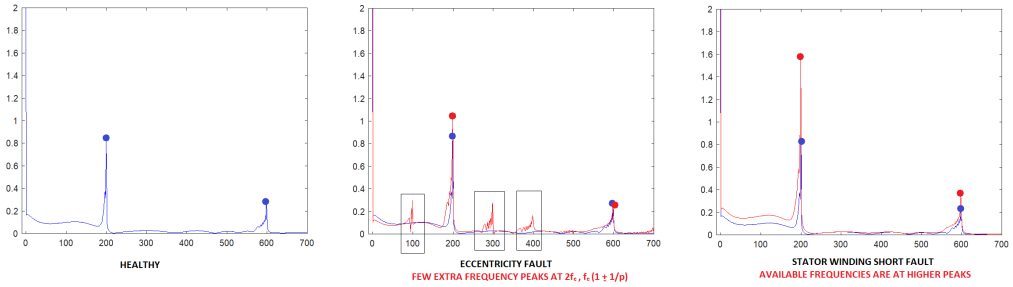


Figure 4.21: 60% of the phase B stator winding shorted, Notice $3f_s, 5f_s, 9f_s$&$13f_s$
(Left to right) Occurrence histogram of frequency ratios obtained from the Wavelet Transform, Fast Fourier Transform
(Top to Bottom) One Speed, One speed (sinusoidal), Three Speed, Three speed (sinusoidal)

In the stator winding faults, the values of frequency to look out for are given by,

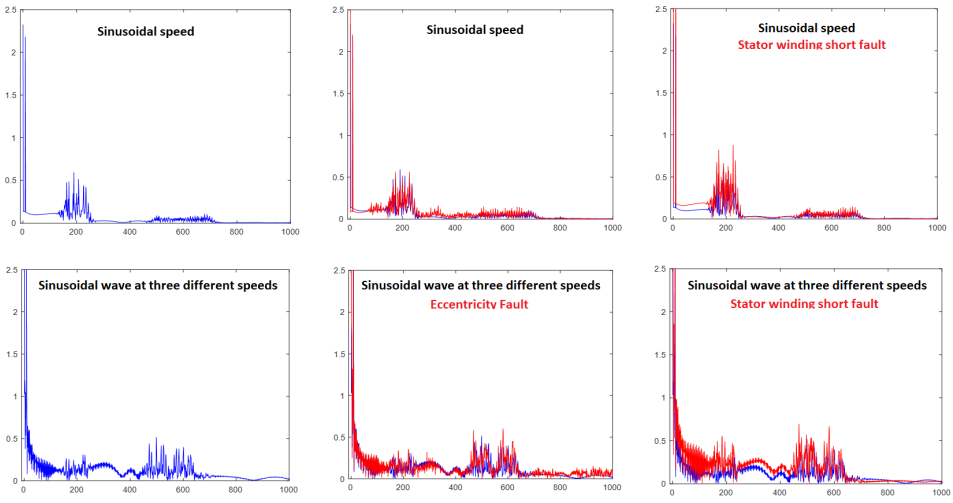
$$f = f_c(2n + 1) \qquad \text{where n=1,2,3...}$$

4.4.1 Constant speed FFTS for constant speed with different faults



FFT does work for when the motor runs at a constant speed. We see how the red faulty signals clearly clears stand out from the healthy motor signals for the same condition. For stator winding fault we get a larger magnitude of the same frequency as more current is drawn in this case and the the mixed eccentricity case we get $0.5f_c$, $1.5f_c$ and $2f_c$

4.4.2 Variable speed FFTS



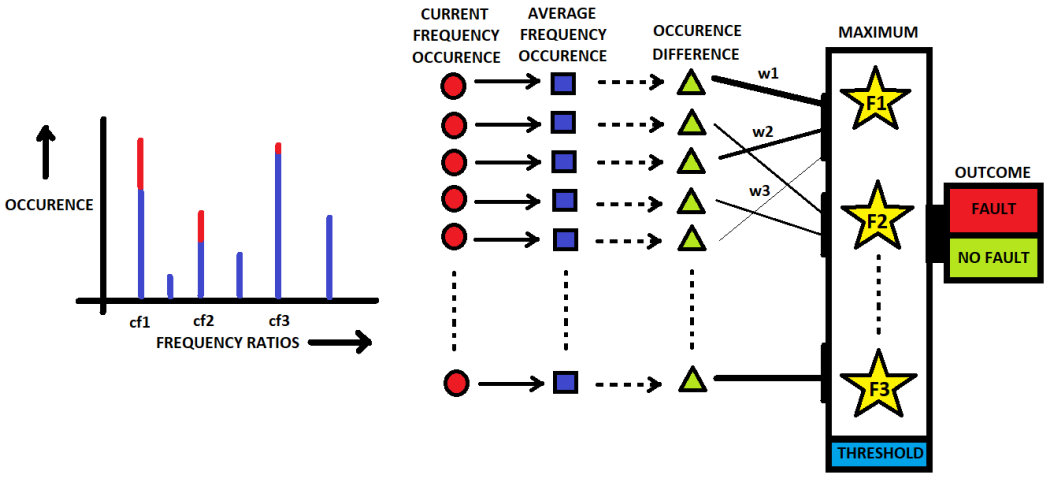
Unlike above, when it is a variable speed signal the FFT loses its capabilities to detection and diagnosis faults. This is due to the absence of a distinct central frequency.

4.5 Utilizing neutral networks as an aid

Using a model-based system on motors has been a challenge for two reasons. Firstly, all motors have a slightly different dynamics. The current gathered from two different PM BLDC motors performing the same action, may have a different frequency spectrum. Though these frequencies will lie within the set of frequencies predicted by MCSA, knowing the exact frequencies to consider as a characteristic of the fault is essential for diagnosis. While the model-based technique tells you where to look for it is a good idea to also have a neural network trained over time to observe the recurring patterns.

Secondly, it is difficult and uneconomical to account for all the small details involved in the system. A simple model is always preferred. As a result, a normal behaviour may get categorized as an abnormality. Attaching a training method may help the system learn from its shortcomings and makes it more robust overtime. It is always better to use them together, as the prediction of the model-based technique can be improved over time by using neural networks.

This section introduces one such technique which can be utilized and improved to diagnose faults in motors.



Faulty frequencies have been derived and used multiple times to monitor the current signal through motor current signature analysis. The possible frequency ratios (FR) due to all the faults are calculated and chosen for investigation like you can see cf1, cf2 and cf3 in the graph. The occurrence of these FRs are estimated over time and compared to normal trends. Like we see in the graph the current occurrence of FRs far exceeds the normal trends at cf1, cf2 and cf3. These difference are sent to a block where its decided which fault is most likely or is there no fault at all (below threshold).

These difference in occurrence of frequency ratios are weighted are sent to the block where they are summed up. We see how $w1 > w2 > w3$ in the figure given above. Each star corresponds to a fault and hence only a selected set of FRs are relevant to each fault (star). The weights are set to unit at first. Then the weight is replaced by the ratio of difference of occurrence but the sum of differences of occurrences of the FRs of that set.

This algorithm to decide the weights can be assigned to a neural network that learns over time. Also, there a threshold maintained at first to differentiate a normal case from faulty cases. This can be set initially to some low value and neutral networks can decide on an optimum value for it over time.

One major drawback with using just neural networks to train is that there is no starting point. The network has to be passed a lot of data to come up with a stable model. With robots inducing a fault can be expensive and time consuming. One huge problem with performing FDD through data analytical means is that they don't get enough fault related data as compared to normal data. This brings a bias into their training. The model-based technique provides a way out of this time consuming dilemma. The first column of the fig 4.15 and 4.16 implement the similar algorithm without neural network but for just two faults. It has no threshold value set though. The same idea is explained better in the future work section.

Chapter 5

Application of idea to KUKA

5.1 Aim and Attempts

As robots take up the dangerous, difficult and mundane jobs of humans in the automotive industry, predictive maintenance is becoming increasing important. While robots can be trusted to increase the productivity of the manufacturing industry, unlike humans they can go out of order all of a sudden due to performing in spite of an internal fault. So, it is important to maintain a robot's health to avoid extra expenses and downtime. KUKA Austin was working towards improving its predictive maintenance toolbox and hence I was offered the opportunity to contribute to this endeavour.

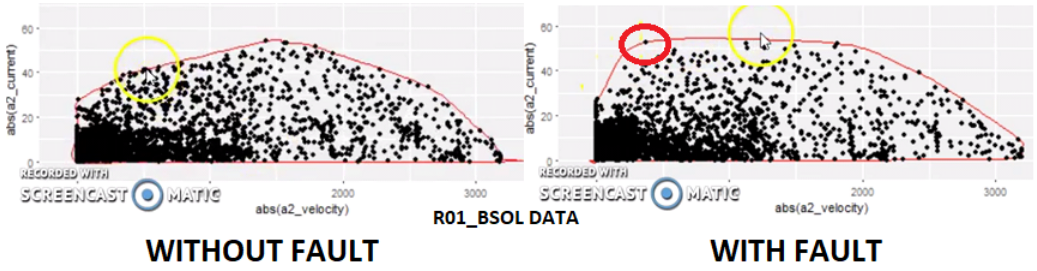
First attempt. The first approach adopted was inspired by my project done earlier in 2018. Here a SCARA robot was modelled in MATLAB Simulink, and it was restricted to follow a predetermined trajectory. The model logged physical quantities (Current, Velocity and Torque) of the dynamic system over time for all the axes. Six individual faults were induced into the system. These faults involved: high friction, increase in stator resistance, external vibration, overload on arm, vibration and loose fittings. Considering the fact that the model parameters and perfect path

taking were known only a simple comparison technique needed to detect a fault (deviation from normal trend). A correlation function sufficed to accomplish the job.

This approach could detect a fault because the profile of a faulty signal was known. So, my first attempt involved me trying to model a KUKA robot which soon proved to be cumbersome due to the numerous intricacies to account for. Considering the trouble to model one robot, it seemed superfluous to think of being able to model all the KUKA robots.

Second attempt. This encouraged me to approach the problem on a microscale than a macroscale. To find small but basic parts of the robot that is most accustomed to failure. Through a thorough literature review about robots, it is evident that most of the fault in robots originate at the motor driver, mechanical coupling mechanisms (belts and gear) or external conditions. Majority of the faults are associated with the motor driver (actuator).

With a more targeted outlook, and a good idea of the resources available to perform predictive maintenance the second attempt concentrated on the idea that every motor has a bunch of characteristics curves within which it performs. A motor that gives outputs outside this region is bound to have an abnormality. So, in order to get speed-torque curves, I utilized the voltage and current data hoping for them to be linearly related. As the motor wasn't running at a constant speed the plot of current vs voltage formed a '8' shaped structure in the four quadrants. As expected during a fault an outlier was generated. Though successful this method threw no light on how to diagnosis the fault. Also, there was no theory supporting its abilities to detect all or most of the motor faults.



Third attempt and discovery. The drive to detect as well as diagnosis faults naturally led me to discovery of frequency methods that have been used extensively in the past and even now to perform FDD. Fourier transform is one of the most well-known techniques widely used in these papers, but it is bound to fail as we are dealing with non-stationary signals (variable speed). More reading guided me to relatively new and rarely used but upcoming technique of wavelet transform. A transform that always a visualization both in frequency and time.

In spite of its astounding capabilities the patterns of the robots motor current in the frequency domain was too vague to draw any conclusion. So visualization was not enough. A better and more automated technique was needed to pick up an abnormality. This brought me to the idea of 'frequency ratios'.

Frequency Ratios. We notice in all the papers written on motor fault detection and diagnosis through Motor Current Signature Analysis (MCSA), they provide a frequency that sprouts up in the presence of a fault. This frequency is always a function of the central frequency. So, frequency ratio is a word I termed to convey fault frequency divided by the central frequency. Frequency ratio is hence independent of the central frequency and is a constant that is a function of the motor parameters. Frequency ratios are more important than frequencies as frequencies change with motor speed but frequency ratios don't.

So the hypothesis adopted in this thesis is that whenever a fault occurs , a new frequency ratio sprouts up or the occurrence of a present frequency ratio grows. This hypothesis yielded really good results but one huge draw back was the low

frequency at which data was accumulated. A lot of essential faulty frequencies get filtered out with such a low collection frequency. This technique could not be tested to perform successful diagnosis , as motor parameters were hard to trace back to and also unavailable to maintain data privacy.

This technique was tested on five previously recorded failures and also more failure cases that were generated at Ausberg to have a good faulty data training set. I have briefly presented these failures and my result on them below.

The five previously recorded datasets where robot failures that occurred in the past. These datasets were sent for analysis by the technicians at the industry. They described the abnormalities in the robot's function that they noticed and listed their actions and the following consequences. When I ran my algorithm to pick up abnormal frequency ratios at the time when a fault occurred it worked successfully. There was a bit of tweaking required to determine a threshold to eliminate noise and pick up the right number of peaks to capture the faults.

I ran the same algorithm Ausberg data wherein half got detected and half failed. This may be because a few faults induced may not have been motor related. Also, my technique was not refined enough to pull out differences in the peaks of occurrence at this stage. The successful results are explained below.

Though this technique worked, the possibility of it being a bad exception stirred me to prove the same results theoretically. So, I decided to model a brushless DC motor from scratch and induce faults into it. And then perform the same tests to see if I get the expected frequency. Though the order of research is reversed in the thesis I believe it was a better idea to refer back to it rather than start with it.

5.2 Three cases: Description with results where this worked

Lets briefly explain the graphs to understand it better. These graphs have experiment number on the x axis and frequency ratios on the y axis. You will notice a number of frequency ratios that occur in one trial and these are denoted by circles whose radius is proportional to its occurrence in that trial.In a way you can imagine this as multiple occurrence histograms(that we saw above) placed side by side and viewed from top. Larger the dots, larger the occurrence of that frequency ratio.

5.2.1 CASE 1

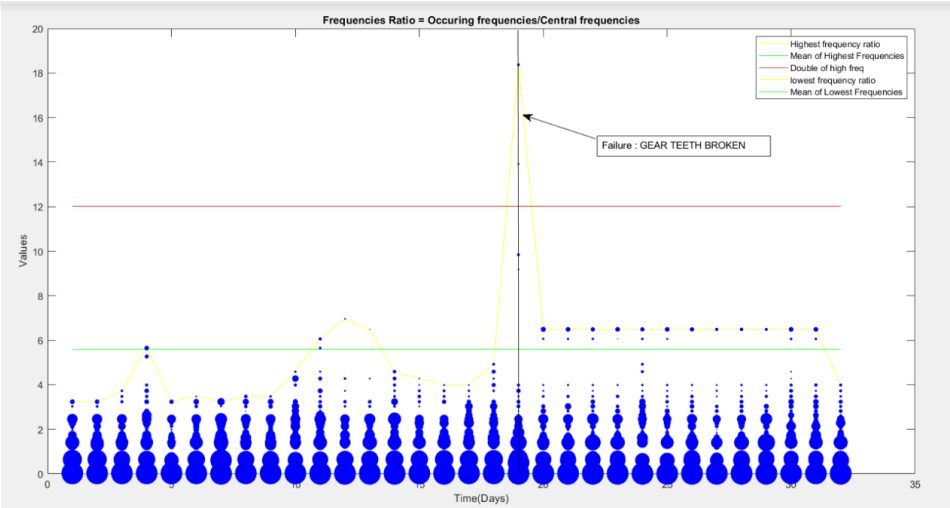


Figure 5.1: Frequency ratios results, peak occurs on failure day

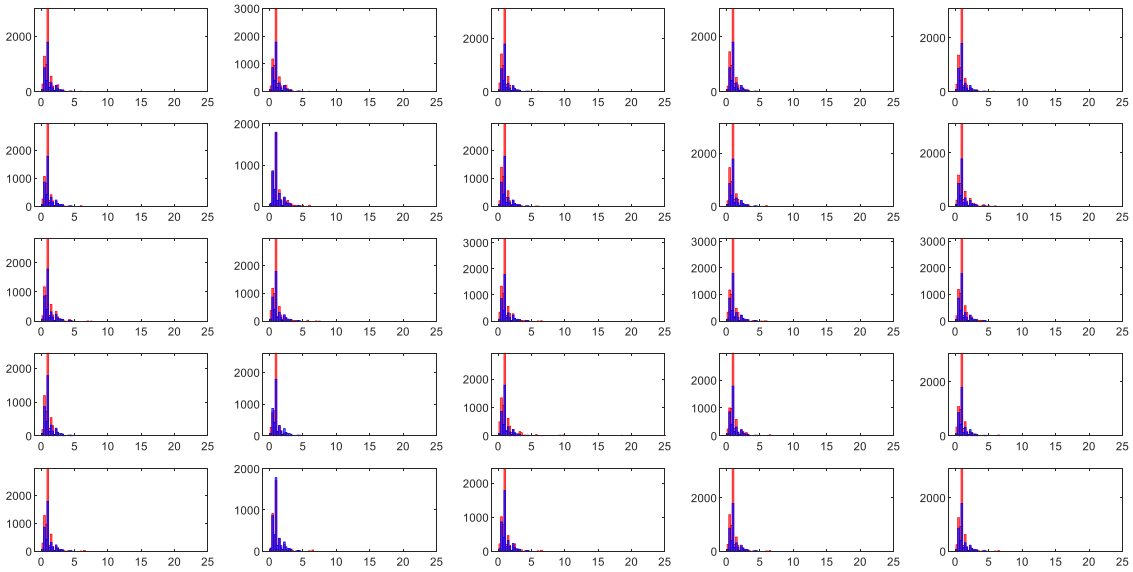
Observation by technician

The gearbox on axis A2 failed on 1/28/2017.

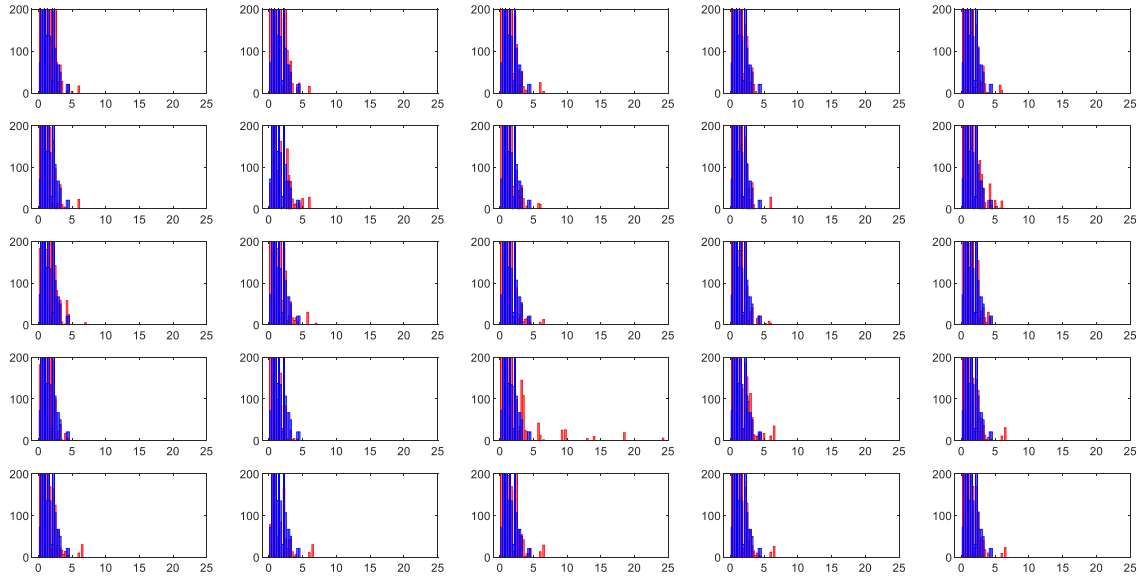
There are many traces taken over roughly 2 years leading up to the failure.

At least one taken on 1/28 after the failure occurred, and many taken since repairs were completed.

Histograms of trials compared to the first trial



Zoomed (look at the 17th window)



The 17th window shows the test case captured on 28th January 2017, the day the motor failed due to gear tooth failure. We notice a few new frequency ratios coming up during this time i.e. at 9, 9.5, 13, 14, 18.5 and 24.5. The threshold is maintained at 35% of the maximum coefficient calculated. It is known that for a gear tooth failure the frequency added to the system is given by:

$$f_{\text{gear fault}} = f_{\text{central frequency}} + k\, f_{\text{driver/driven gear}} \quad \text{, where } k=1, 2, 3\ldots\ldots$$

We are not provided with that information and hence can' verify the formula. The standing out frequency ratio certainly demonstrate that there is a fault. But this technique helps diagnosis the fault as well making model-based methods of FDD more useful than purely data driven method. As mentioned before, an intelligent algorithm can be set up to look at the frequency ratios and estimate which fault is present.

Inferences

In the case given above, we notice that a few new frequency ratios come up at the 17 trial. This trial data was collected on 28th January 2019. We also see a few extra frequencies before too as mentioned in the explanations given by the technicians. That a lot of repairs lead to the final failure. One thing we don't account for here or in any other case is how much does the occurrence changes(radius change). Also mention that they collected a lot of traces even after the failure and we notice a whole new band of frequency ratio after the failure spike.

5.2.2 CASE 2

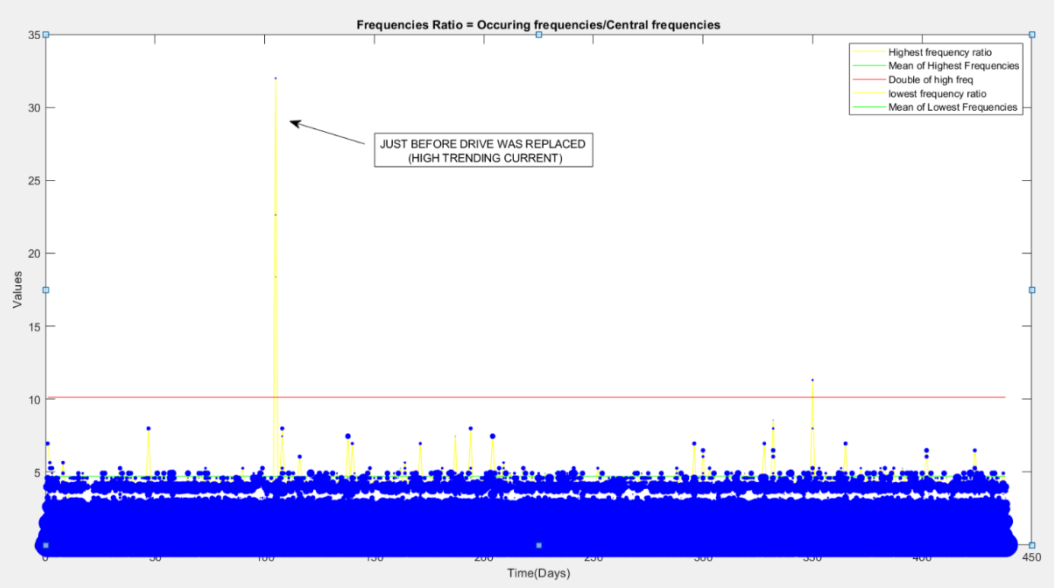


Figure 5.2: Frequency ratio results, peak occurs on failure day

Observation by technician

Early May 2017: A2 max current started to trend high.

On 5/10/2017, maintenance was sent to take a look at this robot.

On 5/12/2017, the counterbalance was greased.

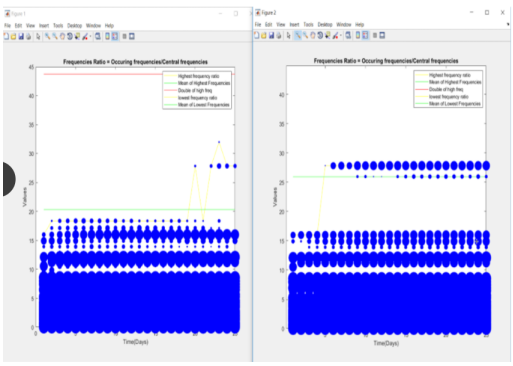
Between 8 and 10 pm on 5/15/2017, the drive was replaced. This brought the max current back down to historical norms.

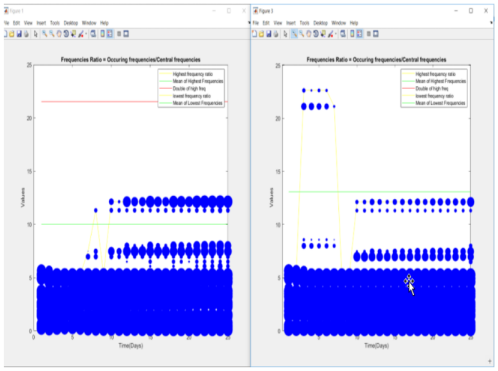
Inferences

The large peak that we notice corresponds to 15th May 2017. The current could be trending high because of a stator winding fault. The assumption of the technicians that the high current is due to high friction and hence greasing didnt solve the problem. But it makes perfect sense that it may have been a motor fault as all got back to normal once the driver was replaced.

We see better results in the case of the theoritical case as I was familiar with the fact that that representation conveys more than these. You can imagine also the frequencies with a larger circle as taller towers like in the histogram. Another drawback is that as the circles overlap with the adjacent frequencies a lot of interpretation we can derive from the graph is lost. This also helped me choose lesser peaks in the theoretical wavelet transform function as essentially the first few peaks are more than sufficients. So I brought the number down to 6.

5.2.3 CASE 3





Observation by technician

Intentional induction of faults at Ausberg.

Aim: To obtain faulty datasets with known faults.

Around 6 programs where run on a robot.

Once without the defect and once with the defect.

Inferences

The right one is defected and left once is normal. We do notice a band of frequency ratios standing out in the faulty case as compared to the normal case. This does convey the presence of a fault.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis is centered around developing a useful motor-based technique to perform fault detection and diagnosis of a robotic motor in non-stationary conditions. It has been majorly inspired by a study conducted at KUKA Austin, a company that performs conditional monitoring of KUKA robots all over the world through cloud computing. My study was dedicated towards developing an effective predictive maintenance tool for the robots. My technique of utilizing frequency ratios as explained elaborately above successfully detected the faulty cases that were offered as a testing dataset. The algorithm developed could be utilized for fault diagnosis apart from detection once the system parameters are known. The next step was to provide theoretical support to my hypothesis. The thesis describes in detail the modelling of a Brushless DC Motor in MATLAB with two inducedfaults: stator inter-turn fault, and eccentricity fault. The same technique is applied to provide supporting results as derived at KUKA, hence verifying the hypothesis. The thesis also encourages using a model based technique for robots whose trajectories are likely to change giving the task needed to be performed.

Characteristics of a model-based technique:

1) Little or no training required: Can work with a new robot model immediately

2) Data Storage : Storing large data to train is not needed

3) Need to make some assumptions: All the complexities of the system cannot be covered through the model.

4) Knowledge in physics: Need an expert to come up with the model

5) System parameters: Need to know the value of a few system parameters to perform FDD in the frequency domain

6) No waste in accumulating data: As we know where to look for to detect a fault, extra data is not logged, saving a lot of infrastructure, energy and work

Characteristics of a data-driven technique:

1) Untrained senarios: Can't detect a defect if not trained with that type of dataset

2) Biasing: The training dataset must be well balanced. It must contain equal proportions of all the possible situations to avoid biasing. The training dataset should have equal proportions of data corresponding to all circumstances to obtain an unbiased result.

3) Independent of system: Needs no knowledge about the system

4) System parameters: System parameters not needed

6.2 Future work

6.2.1 Suggested product design

The most useful method adopted to carry out FDD is the model-based approach assisted with AI techniques [12] [27] [28]. It stands out as most effective as it uses the advantages of both the methods. While the model-based approach is precise in picking faults, independent of the stationarity condition, when a near perfect

model is known, it fails due to the numerous complexities involved that are difficult to account for. Also, it is difficult to assign a weight to the effects of each uncertainty present in the system and also can prove computationally cumbersome. AI techniques on the other hand are unmatched in their capability in finding recurrent patterns in any systems and assigning a model as needed. But AI techniques do have the drawback of needing extensive training to produce a good model and also may collapse if a few conditions of the system change.

The model-plus-AI methods give us the capability of sailing over the disadvantages of each individual approach. This helps to carry out FDD in the most efficient manner. The idea suggested in the thesis is a technique that 'bins' the relevant frequency ratios and weighs the results to estimate which fault is the most probable. This technique does require the knowledge of the system parameters. Also, while this can be applied immediately to a system, training the dataset to use AI techniques makes the approach more robust. The thresholds required to make a decision of whether or not a fault is present and maintenance is required can be made better using AI. These thresholds are decided roughly at the beginning but gets better at detection and diagnosis with more data. So, in a way, while the model takes care of the simple characteristics, AI helps to account for the complexities.

Let's look at the most prominent mechanical failures of a motor and the mechanical components attached to it. These account for the most majority of recorded cases: a) bearing Failure, b) stator inter-turn fault, c) eccentricity (static, dynamic and mixed), and d) gear tooth damage.

While using Motor Current Signature Analysis [31] to detect a fault in a motor working at a constant speed, these are the frequencies we need to look at for

Bearing Fault(Specific case of outer rim defect) [9][7][8] [30]:

$$f_{bf} = f_c \pm k f_{outer} \tag{6.1}$$

$$f_{outer} = \frac{N_{ball}}{2} f_{rm} \left(1 - \frac{D_{ball}}{D_{cage}} cos\beta\right) \tag{6.2}$$

Stator Inter-Turn Fault [5][21]

$$f_{sitf} = f_c \left(2k - 1\right) \tag{6.3}$$

Eccentricity Fault [6]

$$f_{ecc} = f_c \left(\frac{1+k}{p}\right) \tag{6.4}$$

Gear Tooth Failure [10]

$$f_{gdriver} = f_c \pm k f_{driver} \qquad f_{gdriven} = f_c \pm k f_{driven} \tag{6.5}$$

$$f_{driver} = \frac{f_c}{p} \qquad f_{driven} = \frac{f_c}{pGR} \tag{6.6}$$

where

f_c , Electrical frequency

k , Integer

N_{ball} , Number of balls in the groove

f_{rm} , Mechanical rotation frequency

D_{ball} , Diameter of the bearing balls

D_{cage} , Average diameter of inner and outer rim of the bearing cage

β , Contact angle between the ball and the cage

p , Number of pole pairs

GR , Gear ratio

From equations 6.1, 6.3, 6.4 and 6.5, we can calculate the frequency ratios utilizing the system parameters as listed above. The frequency ratios being $\left[1+\frac{N_{ball}}{4}\left(1-\frac{D_{ball}}{D_{cage}}cos\beta\right)\right]$, $(2k-1)$,$\left(\frac{1+k}{p}\right)$,$\left(\frac{1}{p}\right)$ and $\left(\frac{1}{pGR}\right)$.

Each frequency ratio is an indicator of the respective fault. So using the technique developed above where:

1) We perform wavelet transform on the stator current to obtain mxn coefficients,

2) Find the first 'p' highest frequency peaks(from n frequencies) in the m intervals of time.

3) Calculate the 'p' frequency ratios for the m intervals of time.

4) And finally bin these frequencies to look for abnormalities.

We can now diagnose the faults by writing a program to display the occurence of each frequency ratio to see what fault is coming up. We see the results of this approach above. Although it is easy here as there are just two cases. The two faults – stator inter-turn fault and eccentricity fault – can be distinguished on the basis of occurence of even multiplies of the central frequency $2f_c, 4f_c, 6f_c$... as that only comes up in the case of eccentricity. Looking at frequency ratios sometimes it is difficult to tell apart faults due to overlapping frequencies. This can be sorted by looking for unique frequencies and also physical attributes (knocking noise when there is a gear tooth fault or peaking high currents where there is a stator winding fault)

For exampl,e if $N_{ball} = 11$, $D_{ball} = 6.3mm$, $D_{cage} = 28.6$,$p = 2$,$\beta = \pi$ and $GR = 22$. We can choose to bin

Bearing outer rim defect: 5.25,9.5

Stator inter-turn fault: 3, 5,7

Eccentricity: 0.5, 1.5,2.5

Gear driver fault: 2, 4 ,6 and
Gear driven fault: 0.92,0.96,1.04,1.08,....

In spite of the theoritical decision made here, this decision doesn't always work as some frequency ratio might not come up distinctly while the others might. So, an algorithm needs to be established to check these frequency ratios exhaustively for a conclusive result. Also, this is where AI techniques can help. They can help choose these relevant frequency ratios for a machine but noticing the trends, thus making the system wiser and more efficient in the process. As a result, AI uses the model as a heuristic to notice patterns in a limited field of choices making the search algorithm take lesser time.

In addition, the frequency ratios algorithm uses the results from condition monitoring from constant motors current outputs. The technique of using frequency ratios also produces a pseudo constant graph but definitely may carry some errors. The advantage of this graph is that it is valid for variable speed motors. Techniques like dimension reduction is an AI technique that helps to deal with non-stationary signals make it really useful, but has two disadvantages. One, the trajectory of the robot must not change. Situations where a trajectory is performed with breaks in between may still work. But the model needs to be trained all over again once the trajectory is changed. Second, distinguishing between faults can be carried out using SVMs, but this will take a while as training is needed.

6.2.2 Recommendation to the industry

Condition monitoring of motors remotely using an online platform and cloud networking like KUKA Connect has truly been a very progressive initiative to make predictive maintenance a reality. In spite of these amazing ideas and groundbreaking research involved, there are a few physical constraints that may hinder the process.

1) Low frequency data: when data is collected at a very low frequency, fault frequencies may be missed since data is aliased, making these techniques fail. Data was collected from a faulty robot at KUKA at a frequency of 50 Hz which is not as bad as other data collected at 1 Hz. The simulation data was sampled at 1000 Hz and was able to give more useful results.

2) Full period data: The full time data is not needed to pick up a fault. Small traces collected from time to time to work as well using this technique. The fault frequency still exists in the smaller samples.

With the aim of making predictive maintenance efficient it is preferable to collect high frequency data for short intervals of time over full time period data at lower frequencies.Collecting high frequency data at the end of a cycle for a brief period of time will contain the signature of any fault that occurrs during the cycle. In fact, this is precisely the approach that has been implemented by Kuka in their final deployment.

Appendix

MATLAB Code

ECCENTRICITY MAIN PROGRAM

```
%%  BLDC Motor with eccentricity

%% Clearing workspace and command windows; Declaring global variables
clear all
clc
clf
global sigs sigd
global Kp Ki  Rs dt dd d
global P Bm Bl Jm Jl lam tf  lamar lambr lamcr


%% System Parameters in SI units

rs      =       2.75;                           % Coil Resistance   = 2.75
Rs      =       rs*eye(3);
Kp      =       0.2;%0.05;                      % Kp                = 0.02
Ki      =       0.1;%0.0025;                    % Ki                = 0.01
Bm      =       0.000001;                       % Rotor resistance  = 1e-6
Bl      =       0.000001;                       % Load resistance   = 1e-6
Jm      =       0.00004;                        % Load inertia      = 4e-6
Jl      =       0.00005;                        % Rotor inertia     = 1e-6
lam     =       0.00862;                        % Flux linkage      = 8.62e-3
P       =       4;                              % Pole pairs        = 4
dd      =       0.003;
d       =       0.00065;


%% Static Eccentricity fault variables
sigs  =  0.6;
sigd  =  0.2;
[lamar,lambr,lamcr]=magneticFluxVariation(sigs,sigd);

if (sigs==0)&&(sigd==0)
    dd=0;
    d=0;
end


%% Initail Conditions of states
la0     =       0;                              % Inductance coilA    (la)
lb0     =       0;                              % Inductance coilB    (lb)
lc0     =       0;                              % Inductance coilC    (lc)
wm0     =       0;                              % Rotor speed         (wm)
th0     =       0;                              % Switching frequency (wr)
ie0     =       0;                              % Integral gain       (ie)
qr0     =       0;                              % Frequency angle     (qr)

%% Solving for state values over time
x0=[la0;lb0;lc0;wm0;th0;ie0;qr0];               % Equilibium point
tf=10;                                          % Final time
dt=0.0001;                                      % Time interval
tspan=[0 tf];
N=floor(tf/dt);                                 % No of steps
[t,x]=rk4fixed('SEBLDCModFunc',tspan,x0,N);     % Solver
N=length(t);
```

```
 %% Outputs
for i=1:N
    y               =   zeros(1,5);
    [xd,y]          =   SEBLDCModFunc(t(i),x(i,:));
    SpeedMot(i)     =   y(1);
    TorqueMot(i)    =   y(2);
    CA(i)           =   y(3);
    CB(i)           =   y(4);
    CC(i)           =   y(5);
    VA(i)           =   y(6);
    VB(i)           =   y(7);
    VC(i)           =   y(8);
    CTLout(i)       =   y(9);
    Error(i)        =   y(10);
end

%% Display
figure(1)
subplot(3,2,1);
plot(t,smoothdata(SpeedMot)/(2*pi),'g',t,SpeedMot/(2*pi),'b');
xlabel('time(secs)');
ylabel('Speed(rev/s)');
title('Motor angular velocity.')

subplot(3,2,2);
plot(t,TorqueMot,'b');
xlabel('time(secs)');
ylabel('Torque(kgm2/s2');
title('Rotor Torque')

subplot(3,2,3);
plot(t,CA,'r',t,CB,'g',t,CC,'b');
xlabel('time(secs)');
ylabel('Current')
title('Stator current');

subplot(3,2,4);
plot(t,VA,'r',t,VB,'g',t,VC,'b');
xlabel('time(secs)');
ylabel('Voltage(V)');
title('Voltage supplied');

subplot(3,2,5)
plot(t,CTLout);
xlabel('time(secs)');
ylabel('Controller Output(V)');
title('Controller Output');

subplot(3,2,6)
plot(t,Error);
xlabel('time(secs)');
ylabel('Error%');
title('Percentage Error');


function[xdot,y]=SEBLDCModFunc(t,x)
```

ECCENTRICITY FUNCTION

```
%% Constant variables
global wreq Kp Ki Rs P e dd d
global Bm Bl Jm Jl lam lamar lambr lamcr

%% States
la      =       x(1);
lb      =       x(2);
lc      =       x(3);
wm      =       x(4);
theta   =       x(5);
ie      =       x(6);
qr      =       x(7);

%% Constant/Variable speed input
wreq = 20*pi;%+4*pi*sin(((9*pi)/5)*t);

 %% Step speed input
% if t<3
%     wreq = 20*pi;%+4*pi*sin(((9*pi)/5)*t);
% else if t<7
%        wreq = 50*pi;%+4*pi*sin(((9*pi)/5)*t);
%        else if t<10
%            wreq = 60*pi;%+4*pi*sin(((9*pi)/5)*t);
%            else
%               wreq = 70*pi+8*pi*sin(((9*pi)/5)*t);;
%            end
%      end
%  end

%% Calculate phase of the current in coil A,B,& C
if qr<0
 th     =       2*pi+rem(qr,2*pi);
else
 th     =       rem(qr,(2*pi));
end

if theta<0
 Tr     =       2*pi+rem(theta,2*pi);
else
 Tr     =       rem(theta,(2*pi));
end

itr     =       round(Tr*100+1);

%% Inductances

L       =      0.011;
M       =      -0.0052;

Lss     =     [L+dd*sin(4*(Tr-(3*pi/24))) M+d*sin(4*(Tr-(3*pi/24)))
M+d*sin(4*(Tr+(pi/24)));...
               M+d*sin(4*(Tr-(3*pi/24))) L+dd*sin(4*(Tr+(pi/24)))
M+d*sin(4*(Tr+(5*pi/24)));...
               M+d*sin(4*(Tr+(pi/24))) M+d*sin(4*(Tr+(5*pi/24)))
L+dd*sin(4*(Tr+(5*pi/24)))];
```

```
lamp    =    [lamar(itr)*sin(qr- (2*pi)/3);  lambr(itr)*sin(qr+(2*pi)/3);
lamcr(itr)*sin(qr)];

%% Required Variables
i       =       inv(Lss)*([la;lb;lc]-lamp);              %d(current)/dt
ia      =       i(1,1);                                  %Current in coilA
ib      =       i(2,1);                                  %Current in coilB
ic      =       i(3,1);                                  %Current in coilC
e       =       (wreq-wm);                               %Error in speed
ConOut  =       Kp*e + Ki*ie;                            %Controller input
V       =       max(min(ConOut,200),0);                %Voltage Magnitude
v       =       V*0.5*(VoltageSource(th)+1);             %3phase voltage
Cdot    =       v - Rs*i ;                               %d(Inductance)/dt

Te      =       (P/2)*lam*(((ic-0.5*ia-0.5*ib)*cos(qr))...%Torque on Rotor
                +((sqrt(3)/2)*(ia-ib)*sin(qr)));

%% State equations
xdot1   =       Cdot(1,1);                                      %d(la)/dt
xdot2   =       Cdot(2,1);                                      %d(lb)/dt
xdot3   =       Cdot(3,1);                                      %d(lc)/dt
xdot4   =       ((Te)-(Bm+Bl)*wm)/(Jm+Jl);                       %d(wm)/dt
xdot5   =       wm;
if abs(ie)>1000                                                 %d(ie)/dt
xdot6   =       0;
else
xdot6   =       e;
end
xdot7   =       wm*(P/2);                                       %d(qr)/dt

xdot = [xdot1;xdot2;xdot3;xdot4;xdot5;xdot6;xdot7];             %Differenials

%% Outputs
y(1,1) =        wm;                                             %Motor speed
y(1,2) =        Te;                                             %Rotor torque
y(1,3) =        i(1,1);                                         %Currents
y(1,4) =        i(2,1);
y(1,5) =        i(3,1);
y(1,6) =        v(1,1);                                         %Voltage
y(1,7) =        v(2,1);
y(1,8) =        v(3,1);
y(1,9) =        ConOut;                                         %Frequency
y(1,10)=        (e/wreq)*100;                                   %Error
```

STATOR SHORT WINDING FAULT MAIN PROGRAM

```
%%  BLDC Motor with stator short winding

%% Clearing workspace and command windows; Declaring global variables
clear all
clc
global Kp Ki  Rs dt InvLss Lss
global L M P Bm Bl Jm Jl lam tf mu Rf

%% Stator fault inducing variables
 mu     =      0.6;           % Amount of wire cut off due to fault(0-1)
 Rf     =      1;             % Fault Resistance (0.0001-fault,1000-nf)
 L      =      0.00843;
 M      =      -0.00367;
 Lss    =      [L M M -(mu*M);M L M -((1-mu)*mu*M + mu*mu*L);M M L -(mu*M);
                -(mu*M) -((1-mu)*mu*M + mu*mu*L) -(mu*M) mu*mu*L];
 InvLss =       inv(Lss);

%% No Eccentricity
sigs  =  0;
sigd  =  0;
[lamar,lambr,lamcr]=magneticFluxVariation(sigs,sigd);

%% System Parameters in SI units
rs     =      2.75;                          % Coil Resistance   = 2.75
Rs     =      [rs 0 0 0;0 rs 0 -mu*rs;0 0 rs 0;0 -mu*rs 0 mu*(rs)+Rf];
Kp     =      0.2;                           % Kp                = 0.02
Ki     =      0.1;                           % Ki                = 0.01
Bm     =      0.000001;                      % Rotor resistance  = 1e-6
Bl     =      0.000001;                      % Load resistance   = 1e-6
Jm     =      0.00004;                       % Load inertia      = 4e-6
Jl     =      0.00005;                       % Rotor inertia     = 1e-6
lam    =      0.00862;                       % Flux linkage      = 8.62e-3
P      =      4;                             % Pole pairs        = 4

%% Initail Conditions of states
la0    =      0;                             % Inductance coilA    (la)
lb0    =      0;                             % Inductance coilB    (lb)
lc0    =      0;                             % Inductance coilC    (lc)
wm0    =      0;                             % Rotor speed         (wm)
th0    =      0;                             % Switching frequency (wr)
ie0    =      0;                             % Integral gain       (ie)
qr0    =      0;                             % Frequency angle     (qr)
lf0    =      0;                             % Fault lambda        (lf)

%% Solving for state values over time
x0=[la0;lb0;lc0;lf0;wm0;th0;ie0;qr0];       % Equilibium point
tf=10;                                       % Final time
dt=0.0001;                                    % Time interval
tspan=[0 tf];
N=floor(tf/dt);                              % No of steps
[t,x]=rk4fixed('SFBLDCModFunc',tspan,x0,N);   % Solver
N=length(t);
```

```
%% Outputs
for i=1:N
    y                =   zeros(1,5);
    [xd,y]           =   SFBLDCModFunc(t(i),x(i,:));
    SpeedMot(i)      =   y(1);
    TorqueMot(i)     =   y(2);
    CA(i)            =   y(3);
    CB(i)            =   y(4);
    CC(i)            =   y(5);
    CF(i)            =   y(6);
    VA(i)            =   y(7);
    VB(i)            =   y(8);
    VC(i)            =   y(9);
    CTLout(i)        =   y(10);
    Error(i)         =   y(11);
end

%% Display
figure(1)
subplot(3,2,1);
plot(t,smoothdata(SpeedMot)/(2*pi),'g',t,SpeedMot/(2*pi),'b');
xlabel('time(secs)');
ylabel('Speed(rev/s)');
title('Motor angular velocity.')

subplot(3,2,2);
plot(t,TorqueMot,'b');
xlabel('time(secs)');
ylabel('Torque(kgm2/s2');
title('Rotor Torque')

subplot(3,2,3);
plot(t,CA,'r',t,CB,'g',t,CC,'b',t,CF,'y');
xlabel('time(secs)');
ylabel('Current')
title('Stator current');

subplot(3,2,4);
plot(t,VA,'r',t,VB,'g',t,VC,'b');
xlabel('time(secs)');
ylabel('Voltage(V)');
title('Voltage supplied');

subplot(3,2,5)
plot(t,CTLout);
xlabel('time(secs)');
ylabel('Controller Output(V)');
title('Controller Output');

subplot(3,2,6)
plot(t,Error);
xlabel('time(secs)');
ylabel('Error%');
title('Percentage Error');
```

STATOR SHORT WINDING FAULT FUNCTION

```
function[xdot,y]=SFBLDCModFunc(t,x)

%% Constant variables
global wreq Kp Ki Rs P e lam InvLss
global Bm Bl Jm Jl mu

%% States
la      =       x(1);
lb      =       x(2);
lc      =       x(3);
lf      =       x(4);
wm      =       x(5);
theta   =       x(6);
ie      =       x(7);
qr      =       x(8);

%% Constant/Variable speed input
% wreq = 20*pi+4*pi*sin(((9*pi)/5)*t);

 %% Step speed input
if t<3
    wreq = 20*pi;+4*pi*sin(((9*pi)/5)*t);
else if t<7
       wreq = 50*pi;+4*pi*sin(((9*pi)/5)*t);
       else if t<10
           wreq = 60*pi;+4*pi*sin(((9*pi)/5)*t);
           else
              wreq = 70*pi+8*pi*sin(((9*pi)/5)*t);
           end
     end
 end

%% Calculate pahse of the current in coil A,B,& C

if qr<0
 th     =       2*pi+rem(qr,2*pi);
else
 th     =       rem(qr,(2*pi));
end

%% Inductances
lamp   =       lam*[sin(qr-(2*pi)/3);sin(qr+(2*pi)/3);...%PerMag fluxlinkage
               sin(qr);-mu*sin(qr+(2*pi)/3)];

%% Required Variables
i       =       InvLss*([la;lb;lc;lf]-lamp);            %d(current)/dt
ia      =       i(1,1);                                 %Current in coilA
ib      =       i(2,1);                                 %Current in coilB
ic      =       i(3,1);                                 %Current in coilC
iF      =       i(4,1);                                 %FaultCurrent
e       =       (wreq-wm);                              %Error in speed
ConOut  =       Kp*e + Ki*ie;                           %Controller input
V       =       max(min(ConOut,200),0);                 %Voltage Magnitude
v       =       V*0.5*(VoltageSource(th)+1);            %3phase voltage
```

```
v(4,1) =      0;
Cdot   =      v - Rs*i ;                                  %d(Inductance)/dt

Te     =      (P/2)*lam*(((ic-0.5*ia-0.5*ib+mu*0.5*iF)*cos(qr))...
              +((sqrt(3)/2)*(ia-ib+mu*iF)*sin(qr)));    %Torque on Rotor

%% State equations
xdot1  =      Cdot(1,1);                                      %d(la)/dt
xdot2  =      Cdot(2,1);                                      %d(lb)/dt
xdot3  =      Cdot(3,1);
xdot4  =      Cdot(4,1);                                      %d(lc)/dt
xdot5  =      ((Te)-(Bm+Bl)*wm)/(Jm+Jl);                      %d(wm)/dt
xdot6  =      wm;
if abs(ie)>100                                                %d(ie)/dt
xdot7  =      0;
else
xdot7  =      e;
end
xdot8  =      wm*(P/2);                                       %d(qr)/dt

xdot = [xdot1;xdot2;xdot3;xdot4;xdot5;xdot6;xdot7;xdot8];%Differenials%%

%% Outputs
y(1,1) =      wm;                                             %Motor speed
y(1,2) =      Te;                                             %Rotor torque
y(1,3) =      i(1,1);                                         %Currents
y(1,4) =      i(2,1);
y(1,5) =      i(3,1);
y(1,6) =      i(4,1);
y(1,7) =      v(1,1);                                         %Voltage
y(1,8) =      v(2,1);
y(1,9) =      v(3,1);
y(1,10)=      ConOut;                                         %Frequency
y(1,11)=      (e/wreq)*100;                                   %Error
```

VoltageSource()

```
function v = VoltageSource(th)
ch=(floor(th/((pi/6))));
switch ch
    case 0
        v1=6*(th/pi);
        v2=-1;
        v3=1;
    case 1
        v1=1;
        v2=-1;
        v3=2-6*(th/pi);
    case 2
        v1=1;
        v2=-1;
        v3=2-6*(th/pi);
    case 3
        v1=1;
        v2=6*(th/pi)-4;
        v3=-1;
    case 4
        v1=1;
        v2=6*(th/pi)-4;
        v3=-1;
    case 5
        v1=6-6*(th/pi);
        v2=1;
        v3=-1;
    case 6
        v1=6-6*(th/pi);
        v2=1;
        v3=-1;
    case 7
        v1=-1;
        v2=1;
        v3=6*(th/pi)-8;
    case 8
        v1=-1;
        v2=1;
        v3=6*(th/pi)-8;
    case 9
        v1=-1;
        v2=10-6*(th/pi);
        v3=1;
    case 10
        v1=-1;
        v2=10-6*(th/pi);
        v3=1;
    case 11
        v1=6*(th/pi)-12;
        v2=-1;
        v3=1;
    case 12
        v1=0;
        v2=-1;
        v3=1;
    otherwise
       disp('here');
end
v=[v1;v2;v3];
end
```

magneticFluxVariation()

```
function[mmfchga,mmfchgb,mmfchgc]=magneticFluxVariation(sigs,sigd)
n     = 123;

%% Winding Function na nb nc
val=[1/3;2/3;1;1;1;1;2/3;1/3;0;0;0;0];
val=[val;val;val;val;val];
F  =zeros(1,150000);
ctr=1;
j=1;
for i=0:1:150000
        F(i+1)=val(ctr);
        if ~(i<=(j*2618))
            ctr=ctr+1;
            j=j+1;
        end
end
mmfoff  = 10472;
off     = 28798;
fulldeg = 62832;
phdiff  = 20944;
na      = n*F(1+(off):(off+fulldeg));
nb      = n*F(1+(off+phdiff):(off+phdiff+fulldeg));
nc      = n*F(1+(off+2*phdiff):(off+(2*phdiff)+fulldeg));
mmf     = 3*F(1+(mmfoff):(mmfoff+fulldeg));
mmf     = [mmf,mmf,mmf,mmf,mmf];

%% Parameters important to find inductance
g0   = 0.0005;
mu0  = 1.25663706e-6;
r    = 0.0201;
l    = 0.01;
lam  = 44.2;
g    = zeros(1,62832);

%% Initializing
tf   = 12.56;
 for thr=0:0.01:tf
     e    = sqrt(sigs^2+sigd^2+2*sigs*sigd*cos(thr));
     G0   = 1/(g0*sqrt(1-e^2));
     if (sigs==0) && (sigd==0)
     G1=0;
     G2=0;
     alp=0;
     else
     G1   = (2/(g0*sqrt(1-e^2)))*((1-sqrt(1-e^2))/(e));
     G2   = (1/(g0*sqrt(1-e^2)))+((1-sqrt(1-e^2))/(e))^2;
     alp  = atan2((sigd*sin(thr)),(sigs+sigd*cos(thr)))
     end
     g    = G0 - G1*cos(((0:1:62831)*0.0001)-alp);

    mmfchga(round(thr*100+1))=lam*mu0*r*l*sum(na.*g)*(1/10000);
    mmfchgb(round(thr*100+1))=lam*mu0*r*l*sum(nb.*g)*(1/10000);
    mmfchgc(round(thr*100+1))=lam*mu0*r*l*sum(nc.*g)*(1/10000);
 end
end
```

FFT ANALYSIS

```
load('tryd.mat')
load('NorOneSpd.mat')
load('De2Se6OneSpd.mat')
load('StaSht6OneSpd.mat')
load('StaSht9OneSpd.mat')
load('NorOneSpdSine.mat')
load('De2Se6OneSpdSine.mat')
load('StaSht6OneSpdSine.mat')
load('StaSht9OneSpdSine.mat')
load('NorThreeSpd.mat')
load('De2Se6ThreeSpd.mat')
load('StaSht6ThreeSpd.mat')
load('StaSht9ThreeSpd.mat')
load('NorThreeSpdSine.mat')
load('De2Se6ThreeSpdSine.mat')
load('StaSht6ThreeSpdSine.mat')
load('StaSht9ThreeSpdSine.mat')

L = 10001;
Fs= 10000;
f = Fs*(0:(L/2))/L;

tryy=sin();
tryyS=StaSht6OneSpdCB;

Yn  = fft(tryy);             %StatorCB SnapStaticCB
Pn2 = abs(Yn/L);
Pn1 = Pn2(1:((L/2)+1));
Pn1(2:end-1) = 2*Pn1(2:end-1);

YS  = fft(tryyS);
PS2 = abs(YS/L);
PS1 = PS2(1:((L/2)+1));
PS1(2:end-1) = 2*PS1(2:end-1);
%
% Ysc  = fft(tryysc);
% Psc2 = abs(Ysc/L);
% Psc1 = Psc2(1:((L/2)+1));
% Psc1(2:end-1) = 2*Psc1(2:end-1);

figure(1)
plot(f,Pn1,'b',f,PS1,'r');%,f,Psc1,'r')

% xlim([0 100])
%  ylim([0 0.3])
```

CWT

```
load('NorOneSpd.mat')
load('De2Se6OneSpd.mat')
load('StaSht6OneSpd.mat')
load('StaSht9OneSpd.mat')
load('NorOneSpdSine.mat')
load('De2Se6OneSpdSine.mat')
load('StaSht6OneSpdSine.mat')
load('StaSht9OneSpdSine.mat')
load('NorThreeSpd.mat')
load('De2Se6ThreeSpd.mat')
load('StaSht6ThreeSpd.mat')
load('StaSht9ThreeSpd.mat')
load('NorThreeSpdSine.mat')
load('De2Se6ThreeSpdSine.mat')
load('StaSht6ThreeSpdSine.mat')
load('StaSht9ThreeSpdSine.mat')

Fs = 10000;
nop= 6;

 for e = 1:2

 if e==1
[Nml,fsnml] = cwt(NorThreeSpdSineCB,Fs);
 else
[Nml,fsnml] = cwt(NorThreeSpdSineCB,Fs);
 end

Nf= size(Nml,1);
N = size(Nml,2);

Nml = abs(Nml);

%% TO FIND PEAKS and SORT FREQUENCIES IN ASCENDING ORDER

DesFP = zeros(nop,N);
for i=1:N
[p,loc]=findpeaks(Nml(1:Nf,i));
 for j=1:nop
    ln=find(p==max(p),1);
    if p(ln)==0
        break
    else
    DesFP(j,i)=fsnml(loc(ln));
    %DesFPval(j,i)=max(p);
    p(ln)=0;
    end
 end
 DesFPR(:,i)=50*(DesFP(:,i)/DesFP(1,i));
end

h = ceil(max(max((DesFPR))));

if e == 1
    D1 = DesFPR;
```

```
    h1 = h;
else
    D2 = DesFPR;
    h2 = h;
end

end

figure(1)
histogram(D2,2*h2,'FaceColor','r','EdgeColor','r');
hold on
histogram(D1,2*h1,'FaceColor','b','EdgeColor','b');
%xlim([-10,900])
title('Comparison between normal(blue) and red(faulty)')
ylabel('Occurance')
xlabel('Central frequency 50Hz (shifted frequencies)')

D1a=D1(1,:);
D2a=D2(1,:);
for temp=2:1:nop
    D1a=[D1a,D1(temp,:)];
    D2a=[D2a,D2(temp,:)];
end

D1am=ceil(max(D1a));
D2am=ceil(max(D2a));

[Nn,nbinn]=histcounts(D1a,D1am);
[Nf,nbinf]=histcounts(D2a,D2am);

EccFbin         =0;
NorEccbin       =0;
StaShtFbin      =0;
NorStaShtbin    =0;

stop=min(900,min(D1am,D2am));

for i=1:1:stop

    %Eccentricity
    if (nbinf(i)>20 && nbinf(i)<30)||(nbinf(i)>70 &&
nbinf(i)<80)||(nbinf(i)>95 && nbinf(i)<105)
        EccFbin       = EccFbin + Nf(i);
        NorEccbin     = NorEccbin + Nn(i);

    else if (nbinf(i)>145 && nbinf(i)<155)||(nbinf(i)>245 &&
nbinf(i)<255)||(nbinf(i)>445 && nbinf(i)<460)||(nbinf(i)>745 && nbinf(i)<755)
        StaShtFbin    = StaShtFbin + Nf(i);
        NorStaShtbin  = NorStaShtbin + Nn(i);
        end
   end

end
% figure(10)
% stem([0;(EccFbin-NorEccbin);(StaShtFbin-NorStaShtbin);0])
% xlabel('Eccentricity            Stator winding short')
% title('Prediction of which fault is present')
```

SELF AND MUTUAL INDUCTANCES

```
sigs=0.6;
sigd=0.2;

n=123;

%% Winding Function na nb nc

val=[1/3;2/3;1;1;1;1;1;2/3;1/3;0;0;0;0];
val=[val;val;val;val;val];
F  =zeros(1,150000);
ctr=1;
j=1;
for i=0:1:150000
        F(i+1)=val(ctr);
        if ~(i<=(j*2618))
            ctr=ctr+1;
            j=j+1;
        end
end
off     = 28798;
fulldeg = 62832;
phdiff  = 20944;
na=n*F(1+(off):(off+fulldeg));
nb=n*F(1+(off+phdiff):(off+phdiff+fulldeg));
nc=n*F(1+(off+2*phdiff):(off+(2*phdiff)+fulldeg));

%% Parameters important to find inductance

g0   = 0.0005;
mu0  = 1.25663706e-6;
r    = 0.0201;
l    = 0.01;

%% Initializing

ctr = 1;
for thr=0:0.1:6.2
     e    = sqrt(sigs^2+sigd^2+2*sigs*sigd*cos(thr));
     G0   = 1/(g0*sqrt(1-e^2));
     if (sigs==0) && (sigd==0)
     G1=0;
     alp=0;
     else
     G1   = (2/(g0*sqrt(1-e^2)))*((1-sqrt(1-e^2))/(e));
     alp  = atan2((sigd*sin(thr)),(sigs+sigd*cos(thr)));
     end
     g    = G0 - G1*cos(((0:1:62831)*0.0001)-alp);
 for i = 1:62832
    Na(i) = na(i)-(1/(2*pi*G0))*sum((ones(1,62832)*0.0001).*g.*na);
    Nb(i) = nb(i)-(1/(2*pi*G0))*sum((ones(1,62832)*0.0001).*g.*nb);
    Nc(i) = nc(i)-(1/(2*pi*G0))*sum((ones(1,62832)*0.0001).*g.*nc);
end
```

```
laa(ctr)=mu0*r*l*sum((ones(1,62832)*0.0001).*na.*Na.*g);
lbb(ctr)=mu0*r*l*sum((ones(1,62832)*0.0001).*nb.*Nb.*g);
lcc(ctr)=mu0*r*l*sum((ones(1,62832)*0.0001).*nc.*Nc.*g);

lab(ctr)=mu0*r*l*sum(ones(1,62832)*0.0001.*nb.*Na.*g);
lac(ctr)=mu0*r*l*sum(ones(1,62832)*0.0001.*nc.*Na.*g);
lcb(ctr)=mu0*r*l*sum(ones(1,62832)*0.0001.*nb.*Nc.*g);
ctr=ctr+1;
end
```

Bibliography

[1] Tanya M. Anandan, *Readying Your Robots and Workforce for Industry 4.0*. 25th May 2017, https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights//content_id/6553

[2] Cesare Fantuzzi, Cristian Secchi, Antonio Visioli, *On the fault detection and isolation of industrial robot manipulator*. University of Brescia (Italy), IFAC Robot Control, Wroclaw, Poland, Volume 36(17), Pages 399-404.2003

[3] Eliahu Khalastchi and Meir Kalech, *On Fault Detection and Diagnosis in Robotic Systems*. ACM Computing Survey Vol 51, No. 1, Article 9 ,2018.

[4] Paul C. Krause and Oleg Wasynczuk, *Electromechanical Motion Devices*. Chapter 7, Pages 319-322, McGraw-Hill Book Company 1989.

[5] Seung-Tae Lee and Jin Hur, *Detection Technique for Stator Inter-Turn Faults in BLDC Motors Based on Third-Harmonic Components of Line Currents*. IEEE Transactions on Industry Applications, Vol. 53, No. 1, January/February 2017.

[6] Jongman Hong, Sang Bin Lee, Christian Kral and Anton Haumer, *Detection of Airgap Eccentricity for Permanent Magnet Synchronous Motors Based on the d-Axis Inductance*. IEEE Transactions on Power Electronics, Vol. 27, No. 5, May 2012.

[7] A. Rezig ; A. N'Diaye ; M. R. Mekideche ; A. Djerdir *Modeling of a permanent-magnet excited synchronous machine with bearing damage*. XXth International Conference on Electrical Machines, Pages 1778 - 1782, 2012.

[8] Christelle Piantsop Mbo'o and Kay Hameyer *Fault Diagnosis of Bearing Damage by Means of the Linear Discriminant Analysis of Stator Current Features From the Frequency*. IEEE transactions on Industry Applications, Vol. 52, No. 5,September/October 2016.

[9] A. Rezig, A. N'Diaye, M.R Mekideche and A. Djerdir *Modelling and Detection of Bearing Faults in Permanent Magnet Synchronous Motors*. IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society, Pages 3855 - 3860, 2014.

[10] S. Rajagopalan, T. G. Habetler, R. G. Harley, T. Sebastian and B. Lequesne, *Current/Voltage Based Detection of Faults in Gears Coupled to Electric Motors*. IEEE International Conference on Electric Machines and Drives, Pages 1780 - 1787, 2005.

[11] D.C. Karnopp and D.L. Margolis and R.C. Rosenberg, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, John Wiley and Sons, Inc.,Hoboken, NJ, 2012

[12] Vito Mario Fico, Antonio Leopoldo Rodríguez Vázquez, María Ángeles Martín Prats and Franco Bernelli-Zazzera *Failure Detection by Signal Similarity Measurement of Brushless DC Motors*. Energies 2019, 12, 1364,9 April 2019.

[13] A. Ghoggal, S. E. Zouzou, A. Aboubou and M. Sahraoui, *A 2D Model of Induction Machine Dedicated to faults Detection : Extension of the Modified Winding Function*. J. Electrical Systems 1-4 (2005): 69-82.

[14] Robi Polikar, *The Wavelet tutorial*. 2nd edition, 1994.

[15] Jun-Kyu Park, Il-Man Seo, and Jin Hur, *Fault Type Detection using Frequency pattern of Stator Current in IPM-type BLDC Motor under Stator Inter- Turn, Dynamic Eccentricity, and Coupled Faults*. School of Electrical Engineering, University of Ulsan, Ulsan, South Korea, IEEE 2013.

[16] Jawad Faiz and Iman Tabatabaei, *Extension of Winding Function Theory for Nonuniform Air Gap in Electric Machinery* IEEE Transaction on Magnetics, Vol. 38, No. 6, November 2002.

[17] Iman Tabatabaei, Jawad Faiz, H. Lesani, and M. T. Nabavi-Razavi. *Modeling and Simulation of a Salient-Pole Synchronous Generator With Dynamic Eccentricity Using Modified Winding Function Theory*. IEEE Transaction on Magnetics, Vol. 40, No. 3, May 2004.

[18] R. M. Pindoriya, S. Rajendran and P. J. Chauhan, *Speed Control of BLDC Motor using PWM Technique*. International Journal of Advance Engineering and Research Development (IJAERD) ETCEE-2014 Issue, March 2014.

[19] M. A. Awadallah and M. M. Morcos. *Stator-Winding Fault Diagnosis of PM Brushless DC Motor Drives*. Proceedings of the 2002 Large Engineering Systems Conference on Power Engineering, 2002 IEEE, 0-7803-7520-3.

[20] Yanxin Li, Qinfen Lu, Z. Q. Zhu, L. J. Wu, G. J. Li and Di Wu1, *Analytical Synthesis of Air-Gap Field Distribution in Permanent Magnet Machines With Rotor Eccentricity by Superposition Method*. IEEE Transaction on Magnetics, Vol. 51, No. 11, November 2015.

[21] Hong Liang, Yong Chen, Siyuan Liang and Chengdong Wang, *Fault Detection of Stator Inter-Turn Short-Circuit in PMSM on Stator Current and Vibration Signal*. Applied Sciences, 8, 1677,16 September 2018.

[22] A. Purna Chandra Rao, Y. P. Obulesh and Ch. Sai Babu *Mathematical Modeling of BLDC Motor with closed loop speed control using PID Controller under various loading conditions*. ARPN Journal of Engineering and Applied Sciences, Vol. 7, No. 10, October 2012.

[23] Qi Li , Tao Fan and Xuhui Wen, *Armature-Reaction Magnetic Field Analysis for Interior Permanent Magnet Motor Based on Winding Function Theory* IEEE Transaction on Magnetics, Vol. 49, No. 3, March 2013.

[24] Bashir Mahdi Ebrahimi, Jawad Faiz and Mehrsan Javan Roshtkhari, *Static-, Dynamic-, and Mixed-Eccentricity Fault Diagnoses in Permanent-Magnet Synchronous Motors*. IEEE Transaction on Industrial Electronics, Vol. 56, No. 11, November 2009.

[25] Ao Zhang ID, Yan Bai, Bo Yang and He Li, *Analysis of Nonlinear Vibration in Permanent Magnet Synchronous Motors under Unbalanced Magnetic Pull*. Applied Sciences, 8, 113,15 January 2018.

[26] Z. J. Liua, J. T. Lib and M. A. Jabbarb, *Prediction and analysis of magnetic forces in permanent magnet brushless dc motor with rotor eccentricity* Journal of Applied Physic 99, 08R321 (2006)

[27] Mehmet Akar, Mahmut Hekim, Umut Orhan *Mechanical fault detection in permanent magnet synchronous motors using equal width discretization-based probability distribution and a neural network model*. Turkish Journal of Electrical Engineering & Computer Sciences, 2015.

[28] Mehrdad Heydarzadeh, Mohsen Zafarani, Enes Ugur, Bilal Akin and Mehrdad Nourani, *A Model-based Signal Processing Method for Fault Diagnosis in PMSM Machine*. IEEE Energy Conversion Congress and Exposition (ECCE) Pages: 3160 - 3164, 2017.

[29] Hossein Hooshmandi, Mohammad Ebrahimi1, Ali Davoudi, and Alireza Pouramin, *Analytical Derivation of Induction Motors Inductances under Eccentricity Conditions*. Electromagnetics Research B, Vol. 60, 95, 15 June 2014.

[30] Mansour Ojaghi and Mahdi Sabouri, *Dynamic Modeling and Simulation of Induction Motors with Different Bearing Faults*. 2015 18th International conference on electric machines and systems (ICEMS), Oct 25-28, Pattaya City, Thailand.IEEE 2015.

[31] G Sreedhar Babu, A Lingamurthy and A S Sekhar, *Condition monitoring of brushless DC motor-based electromechanical linear actuators using motor current signature analysis*. The International Journal of Condition Monitoring ,Volume 1 , Issue 1 , June 2011.

[32] Ibrahem Hussein, Zakariya Al-Hamouz , M. A. Abido and Abdulaziz Milhem, *On the Mathematical Modeling of Line-Start Permanent Magnet Synchronous Motors under Static Eccentricity*. Energies 2018, 11, 197, 16 January 2018.

[33] Jawad Faiz and Siavash Pakdelian, *Finite-Element Analysis of a Switched Reluctance Motor Under Static Eccentricity Fault*. IEEE Transaction of Magnetics, Vol. 42, No. 8, August 2006.

[34] Jawad Ahmed Farooq, Tsarafidy Raminosoa, Abdesslem Djerdir and Abdellatif Miraoui, *Modelling and simulation of stator winding inter-turn faults in permanent magnet synchronous motors*. The international journal for computation and mathematics in electrical and electronic engineering, Vol. 27,Issue: 4, pp.887-896. 2008

[35] Michel Misiti, Yves Misiti, Georges Oppenheim and Jean-Michel Poggi *Wavelet Toolbox, For Use with MATLAB* User's Guide, Version 3, The MathWorks, 1997.

[36] PF Albrecht, JC Appiarius, EP Cornell, DW Houghtaling, RM McCoy, EL Owen, DK Sharma, *Assessment of the Reliability of Motors in Utility Applications*. IEEE Transactions on Energy Conversion, Vol. EC-2, No. 3, September 1987.

[37] Xia, Chang-liang, *Permanent Magnet Brushless DC Motor Drives and Controls*. John Wiley and Sons, pg 18–19, 2012.

[38] Anonymous, *1-3-2 Brushless DC Motor*. https://www.nidec.com/en-EU/technology/motor/basic/00005/ 1995.